
# GRAPH-BASED TEMPORAL ANALYSIS IN DIGITAL FORENSICS

THESIS

Nikolai A. Adderley, 1stLt, USAF

AFIT-ENG-MS-19-M-005

**DEPARTMENT OF THE AIR FORCE**
**AIR UNIVERSITY**

***AIR FORCE INSTITUTE OF TECHNOLOGY***

**Wright-Patterson Air Force Base, Ohio**

AFIT-ENG-MS-19-M-005

# GRAPH-BASED TEMPORAL ANALYSIS IN DIGITAL FORENSICS

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

Nikolai A. Adderley, B.S

1stLt, USAF

March 2019

AFIT-ENG-MS-19-M-005

# GRAPH-BASED TEMPORAL ANALYSIS IN DIGITAL FORENSICS

THESIS

Nikolai A. Adderley, B.S
1stLt, USAF

Committee Membership:

Dr. Gilbert L. Peterson
Chair

Dr. Robert Mills
Member

Dr. Douglas Hodson
Member

AFIT-ENG-MS-19-M-005

# Abstract

Establishing a timeline as part of a digital forensics investigation is a vital part of understanding the order in which system events occurred. However, most digital forensics tools present timelines as histogram or as raw artifacts. Consequently, digital forensics examiners are forced to rely on manual, labor-intensive practices to reconstruct system events. Current digital forensics analysis tools are at their technological limit with the increasing storage and complexity of data. A graph-based timeline can present digital forensics evidence in a structure that can be immediately understood and effortlessly focused. This paper presents the Temporal Analysis Integration Management Application (TAIMA) to enhance digital forensics analysis via information visualization (infovis) techniques. TAIMA is a prototype application that provides a graph-based timeline for event reconstruction using abstraction and visualization techniques. A workflow illustration and pilot usability study provided evidence that TAIMA assisted digital forensics specialists in identifying key system events during digital forensics analysis.

# Acknowledgements

To my loving and supportive wife and children. Thank you for your sacrifice over the last two years and always being in my corner making sure I had all the love and support I needed to get through this.

Additionally, I would like to thank my thesis chair and supervisor Dr. Gilbert L. Peterson for his astute guidance, patience and direction throughout my course work and thesis process.

Moreover, I would also like to thank Capt. Daniel Schelkoph for his efforts in creating a test Neo4j database and assistance in incorporating it into TAIMA.

Nikolai A. Adderley

# Table of Contents

Page

Page

# List of Figures

# List of Tables

GRAPH-BASED TEMPORAL ANALYSIS

IN DIGITAL FORENSICS

# I. Introduction

A significant challenge within the digital forensics community is conducting digital forensics analysis. More specifically, determining which system events occurred and the order in which those events occurred [9]. Over the last 15 years digital forensics has been under relentless pressure as a result of the rapid digital technological developments, increased storage, heterogeneous data and the rise of digital device ubiquity [2]. An additional issue is the increased use of computers in the commission of a crime [10].

Digital forensics in the late 1990s usually included only one computer hard drive loaded with a version of Microsoft Windows [9]. As a result, digital evidence collection rarely exceeded the megabytes threshold. Garfinkel [11] described the era as the *'Golden Age'* of computer forensics. Developed to target mostly one operating system, loaded on hard disk drives known to not exceed the megabytes threshold made the *'first-generation'* [12] digital forensics tools proficient and they quickly become industry standard tools in the digital forensics domain [12]. Compared to other tools at that time, they made certain aspects of digital forensics analysis easier [12], albeit, basic digital forensics analysis processes were still done manually and were labor-intensive [13], [2], [6], [12]. Recently, however, the digital technology development revolution has pushed the *'first generation'* tools to their limits.

The *'first generation'* tools are outdated and suffer from scalability limitations

due to data volume and the complexity of digital evidence produced by modern digital technology. Exasperating the situation further is examiners having to also deal with a wide assortment of what is now considered a 'computer'; such as, mobile devices, watches, fitness trackers and tablets [14]. Additionally, ownership cost of those devices is at an all-time low, resulting in most Americans now owning multiple computers [10]. According to a 2018 Pew Research Center survey, a substantial majority of Americans (75%) own various forms of digital devices, i.e., smartphone, desktop computer or laptop [10]. Consequently, in a criminal investigation, collecting various forms of devices per investigation has become the norm. As the situation worsens, the digital technological gap between digital forensics tools and digital technology appears to be widening with no reliable solutions.

Previous research proposed information visualization (infovis) and abstraction as a solution to address the challenges [15]. Through leveraging a human's perceptual and intellectual capabilities, best practice infovis and abstraction can provide insight into large and complex data by minimizing the adverse effects of *'information overload'* by reduce the amount of data displayed to the user [16]. Hibshi et al. [17] reported digital forensics experts showed appreciation for displays that reduces the number of items for review while still displaying relevant information.

This research proposes a novel infovis application to assist digital forensics analysts in mitigating the effects of *'information overload'*. The proposed application leverages temporal event reconstruction techniques and infovis practices to enrich a graphical timeline to determine the order in which system events occurred and the time those events occurred. The graph-based timeline presents digital forensics evidence in a intuitive web-based platform that reduces the need for manual, labor-intensive digital forensics analysis practices.

## 1.1 Digital Forensics

The digital forensics process is a set of outlined steps that law enforcement, investigators and forensics analysis should follow to ensure legal admissibility of their findings. There have been several digital forensics process models proposed as described in [18]. This research uses the National Institute of Standards and Technology (NIST) [1] definition:

> *'the application of science to the identification, collection, examination, and analysis of data while preserving the integrity of the information and maintaining a strict chain of custody for the data.'*

While the digital forensics community does not have a single process for digital forensics, the community does, however, generally agree that digital forensics has four primary phases [1]:

- **Collection:** Extracting relevant data in a forensically sound manner
- **Examination:** While maintaining data integrity, 'interrogate' the data to
- **Analysis:** investigate the data to answer questions regarding the investigation
- **Reporting:** The results of the analysis is presented in this phase

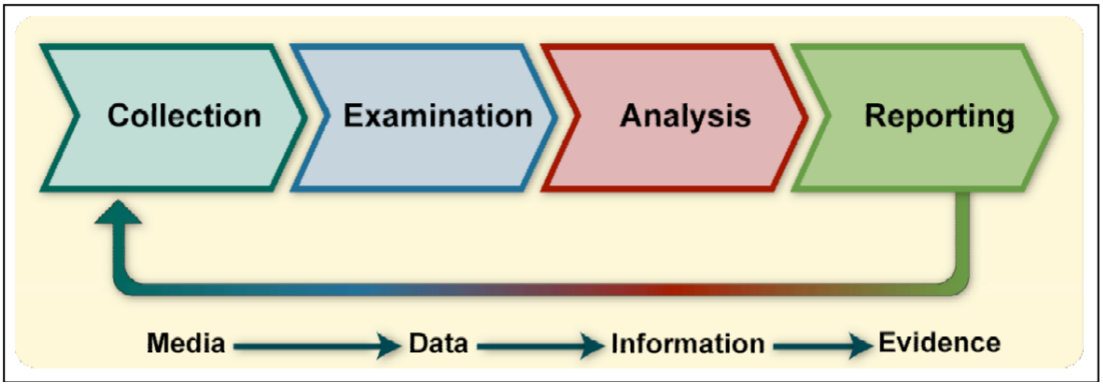



**Figure 1. The Digital Forensics Process [1].**

This research concerns the analysis phase. The Digital Forensics Research Workshop (DFRWS) defines the analysis phase as [19]:

> *"To identify digital evidence using scientifically derived and proven methods that can be used to facilitate or further the reconstruction of events in an investigation."*

As in a routine investigation, digital forensics analysis attempts to discover information to answer the 5W's (Who, What, When, Where, Why) inside the confines of a digital environment. A significant part of the analysis process is event reconstruction; knowing the chronological order of system events [1]. During event reconstruction, a timeline assists the examiner in finding out which user or application created, accessed, modified, received, sent, viewed, deleted, and launched each artifact of interest or when those events occurred and how those artifacts ended up on the device. The Organization of Scientific Area Committees for Forensics Science defines Reconstruction as:

> *"Organize observed traces to disclose the most likely operational conditions or capabilities (functional anal-*

*ysis), patterns in time (temporal analysis), and linkages between entities – people, places, objects – (relational analysis) [20]."*

## 1.2 Problem Statement

The problem to be resolved is how to minimize the manual effort required during a digital forensics analysis caused by data volume size and data complexity (heterogeneous data) [21]. As Hales stated [22], *"The workload for investigators is increasing, and the time required to analyze the datasets is not decreasing to compensate."* Currently, with the explosive evolution of digital technology development and the ubiquity of mobile devices, digital forensics analysis progressively continues to get more complicated [23].

While the digital technology revolution has changed the landscape of modern computing, industry standards tools are still using outdated technology from the 1990s. One major challenge for examiners is the use of text-based displayed [17]. Schrenk and Poisel [24] conducted a comparison study between two text-based display tools and two graph-based visualization tools. In the first comparison, Goodall [25] compared 'Time-based Network Traffic Visualizer' (TNV), a graph-based tool, to Wireshark, a text-based network forensics tool. The results showed that the participants favored 'TNV' over 'Wireshark.' In the second comparison, Olsson and Boldt [6] compared a graph-based infovis tool, 'CyberForensics TimeLab'(CFTL) and 'Forensics Tool Kit' (FTK), a text-based tool. Survey statistics revealed that CFTL was preferred and was faster at solving the case. In both experiments, the group that used the graph-based tools, TNV and CFTL, had more correct answers than the group that used the text-based tools. In summary, the authors claimed industry standard tools like, EnCase, FTK, SleuthKit

or ProDiscover have the technology to conduct an extensive and detailed analysis of forensics data, but lack graph-based timeline and event reconstruction visualization abilities.

This work presents Temporal Analysis Integration Management Application (TAIMA), an infovis application prototype that enhances digital forensics investigations with an emphasis on the analysis phase of the digital forensics process. TAIMA enhances timeline creation and event reconstruction by providing a graphical timeline with temporal abstraction and visualization techniques. The graphical timeline displays a vast amount of heterogeneous data in sequential order based on temporal attributes. The abstraction technique transforms low-level execution traces into high-level events in a way that is understandable and informative. The infovis technique allows the analyst to adjust their focus and attention from a broad case wide overview to a detailed low-level view of digital forensics traces. The engineering produces a graphical timeline with exploration capabilities at the examiner's fingertips.

As part of the research, a usability study provided participants with access to TAIMA to complete a simulated digital forensics analysis task. All participants completed the task and gave TAIMA an overall satisfaction rating. Furthermore, a demonstration described abstraction techniques used to minimize the number of items on the display which led to the rapid discovery of suspicious files.

### 1.3 Research Hypotheses

The research hypothesis is that an interactive GUI with a graph-based timeline can minimize the challenges of digital forensics analysis. Previous studies show that integrating advanced infovis methods and practices to digital forensics tools reduce investigative timeline and significantly increase accuracy in discovering

relevant digital evidence [22],[6],[26],[27]. Infovis modus operandi infuse graphics into an interactive digital environment to support comprehension of complex of data [13]. As a result, relationships and data patterns that might not be identified in a text-based display can be uncovered and recognized easier with graph-based displays integrated with best practice infovis.

### 1.4 Research Questions

The overall goal of this research is to improve digital forensics analysis via a graphical timeline and visual analysis by; (1) demonstrating efficiency infovis and abstraction methods and practices that reduces the digital forensics challenges caused by digital evidence volume and complexity; and (2) proving effectiveness by evaluating results of the pilot usability test. To help achieve the research goal a review of the literature provided guidance on how to best utilize technology and infovis techniques for use in a graphical user interface (GUI). The theoretical knowledge gained from the literature review helped formed the following question that guided this research effort:

*(1) What Information Visualization (infovis) practices reduce the digital forensics challenges of evidence volume and complexity within the digital forensics analysis process?*

The answer to this question comes from a comprehensive review of existing literature and personal experience. The review provides theoretical knowledge of infovis practices that successfully mitigate the challenges.

*(2) To what degree does the use of a graphical timeline integrated with information Visualization best-practices support the digital analysis process?*

The answer to this question stems from the analysis of the pilot usability test results and abstraction technique evaluation. The usability test evaluation measures the participant's inclination toward the research hypothesis using the arithmetic mean. The abstraction technique evaluation measures the efficiency of the technique to reduce the dataset while still displaying relevant information.

## 1.5 Contributions

This work contributes to the body of knowledge by providing evidence of the effects of a graphical timeline in supporting digital forensics analysis and digital event reconstruction. The potential impact of this research is a reduction on the reliance on manual processes during a forensics analysis. The novel software engineering techniques and use of state-of-the-art technology provided solutions to the challenges caused by increased data and volume and data complexity.

The application of previous works, exploration of proven theories, tactics and techniques, supported with empirical results expands the digital forensics domain literature. As a result, future practitioners now have additional resources and solutions for implementing effective software engineering practices in their attempts to develop solutions to mitigate digital forensics challenges.

Furthermore, this research leverages modern technologies of state-of-art-technologies and improved methodologies to digital forensics through the use of the GRAND-Stack (GraphQL, React, Apollo and Neo4j Database) [28]. The stack is written in JavaScript which is widely accepted for providing dynamic visualization for an interactive User Interface and User Experience in web browsers [29].

## 1.6 Organization of the Thesis

The remainder of this paper is structured as follows: Section 2 discusses background information and provides justification that supports the need for a graph-based timeline approach in a digital forensics investigation. Section 3 describes TAIMA and how it uses GRANStack to transform data for visual display on a graph-based timeline. Chapter 4 discusses the research methodology design. Chapter 5 discusses the results of the usability and case study, while Chapter 6 concludes the thesis and recommends future work and improvement to the prototype.

# II. Literature Review

*"No single tool or technique yet provides the analyst with the means to vary the focus of their attention from low-level detail to case-wide overview nor provides the means to organize evidence into reconstruction of activity by linking related/correlated low-level data items." -Hales[22]*

The rapid increase in digital technology over the last decade has created significant challenges for digital forensics analysts and have shaped how forensics professionals execute digital forensics investigations. In addition to the proliferation of digital devices, digital forensics analysts are struggling with modern devices advanced technology, larger storage capacity and the vast amount of heterogeneous data [13]. The FBI now classifies 'digital devices' as desktop computers, laptops, mobile devices (cell phones and tablets), GPS navigation devices, vehicle computer systems, Internet of Things (IoT) devices, and much more [14]. Currently, the examination of one hard disk hard is a manageable task; however, multiple modern hard disk drives with vast storage capacity might be impossible. The community needs innovative tools and capabilities as the limited number of specialists continues to shrink due to the inability to hire more and tools continue to be outdated.

An area for innovation comes from information visualization (infovis) [30]. However, the amount of previous work focusing on visualization and forensic analysis procedures in digital forensics is limited [6], [22], [31]. In 2011, Carbone [32] compiled and assessed a comprehensive list of established digital forensics applications with digital timeline generation capabilities. The report revealed a majority of the tools had limited timeline visualization capabilities, or lack the capability altogether. More importantly, the author acknowledged that *"no intuitive GUI-*

*based timeline visualization software yet exists due primarily to the difficulty in developing an application capable of responding to the needs of investigators when dealing with large datasets."*

This thesis presents an intuitive graph-based timeline visualization prototype application developed to minimize the challenges of large datasets by integrating infovis technology into a graphical user interface (GUI). The GUI uses a graph back-end database as storage and renders a infovis that reduces large datasets while still providing a global perspective on the vast digital artifacts dataset on a graphical timeline.

This chapter surveys previous works related to various developments in infovis, abstraction, temporal analysis, digital forensics timelines and Human-Computer Interaction (HCI) and usability testing. The first section establishes a working definition of infovis and discusses the three influential infovis frameworks for this research. Section two examines abstraction techniques used to develop a reduction strategy for vast datasets. Section three outlines temporal analysis. Section four studies digital forensics timelines strengths and weaknesses. The fifth and final section investigates the importance of Human Computing Interfaces (HCI) experiments, specifically, examining controlled experiments and usability testing to evaluate the GUI.

## 2.1 Information Visualization (infovis) Frameworks

This research uses Card, et al. [33] definition of infovis:

> *"The use of computer-supported, interactive, visual representations of abstract data to amplify cognition."*

The definition amplifies the significance of harnessing computing power technology to display large volumes of data to examiners in an intuitive manner for

rapid examination and analysis. Shneiderman [34] expressed support for infovis by stating, *"The eye, the hand, and the mind seem to work smoothly and rapidly as users perform action on visual displays."* Previous work contains empirical studies that confirm infovis enhances visual analysis and accelerate digital evidence detection by leveraging the human senses of the mind and eyes [30], [22]. According to a report on infographics [35], 90% of the data transmitted to the brain is visual. Moreover, people can process visual information exponentially faster than text [35].

Infovis has seen an increase in research and development as developers explore how to harness the processing power and graphics of the modern computers [36]. Similarly, Osborne and Slay [30] explain how infovis takes advantage of the eyes and mind in a way that addresses the *'information overload'* faced by examiners. Moreover, the article credited infovis for empowering both trained examiners and the layman alike to be effective *'data detectives.'* Lastly, in their report, Pati et al. [15] argues that visualization techniques are noteworthy because it effectively takes advantage of graphics to convey information.

The next section discusses three infovis models that influenced this thesis: (1) The Visual Information Seeking Mantra, (2) The Explore, Investigate and Correlate (EIC) Conceptual Framework and (3) The Visualization Pipeline. A review of the three models provided theoretical knowledge of techniques, models and practices for effective timeline visualization rendering strategies and best practices data transformation model.

### 2.1.1 The Visual Information Seeking Mantra.

Shneiderman [34] proposed a GUI design guide referred to as the Visual Information-Seeking Mantra: overview first, zoom and filter, then details on de-

mand. Renowned in the infovis community, the mantra simplifies GUI design by mapping complex GUI design capabilities to just seven critical tasks. When implemented correctly throughout the GUI design mantra inspired GUIs enable examiners to rapidly identify correlations between system events via an overview perspective that reduces the amount of data via abstraction techniques [23]. Additionally, strict adherence to Mantra leads to fast loading displays and responsive user-controlled exploratory capabilities [34].

Shneiderman proposes a task by data type taxonomy with seven data types and seven tasks.

***Seven Data Types***

1. One-dimensional data
2. Two-dimensional data
3. Three-dimensional data
4. Temporal data
5. Multidimensional data
6. Tree data
7. Network Data

***Seven Tasks***

1. Overview
2. Zoom
3. Filter
4. Details-on-demand
5. Relate
6. History
7. Extract

Schneiderman further argues the *'golden age'* of building custom-made application to process only one data type have passed. Modern effective and efficient infovis applications must be able to process more than one data type and offer all seven capabilities listed in the Seven Tasks. Additionally, an important consideration for modern application is to integrate with other mainstream digital forensics applications [34].

### 2.1.2 The Explore, Investigate and Correlate (EIC) Conceptual Framework.

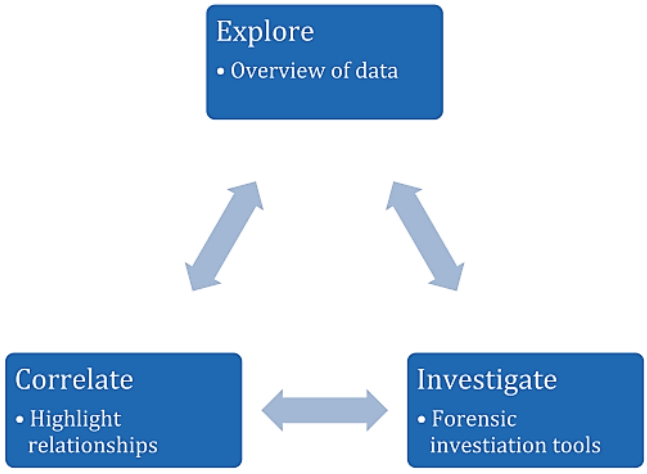



**Figure 2. The Three Phases of the EIC Process [2].**

Osborne et al. [2] were the first to apply the infovis mantra to digital forensics. The authors developed a high-level conceptual framework for digital forensics referred to as 'Explore, Investigate and Correlate' (EIC) process. The goal was to contribute to the field by streamlining infovis processes to make digital forensics analysis less labor intensive. The study highlighted the difficulties in finding digital forensics applications that can scale and are efficient solutions for displaying and analyzing a large volume of information. In particular, the authors highlighted the lack of research regarding formal digital forensics processes and frameworks

and emphasized the need to develop solutions to mitigate the challenges in the digital forensics domain.

The EIC framework was designed to mitigate some of the critical digital forensics domain challenges by integrating the mantra mentality: Overview first, zoom and filter, then details-on-demand. The authors implemented the mantra mentality as part of the EIC framework to provide common visualization controls to users. Like the mantra, the EIC process also has three parts. After the examiner has 'explored' the overview and applied filters to identify files of interest, the investigation moves into the 'investigate' phase. In this phase, the examiners begin to examine suspicious files in more detail to identify 'correlation' among the artifacts.

The authors developed a web-based proof of concept implementation of the EIC process. Aimed at scalability and volume, the system was designed to help examiners get a clearer understanding of a vast amount of data using infovis techniques combined with user-directed exploration functionalities. A user study involving participants from the digital forensics and intelligence domains demonstrated that the EIC process assists in analyzing vast amount of digital evidence.

### 2.1.3 Visualization Pipeline.

While the EIC framework supports the Analysis and Presentation phases of the Digital Forensics Process, it does not address transforming raw data into a visual state. For that this research examined the Visualization Pipeline.

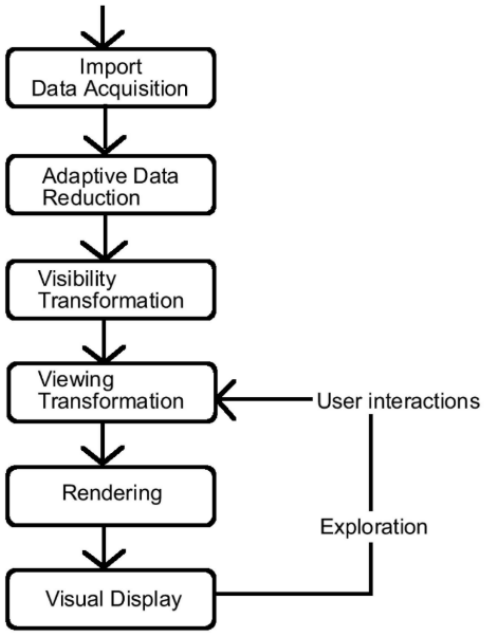



**Figure 3. The Visualization Pipeline [3].**

The Visualization Pipeline is a foundational model for transforming data from its raw state into a structure to use in a visual display. Groth [3] used the visualization pipeline to develop the Knowledge Discovery in Database (KDD) process that centers around user interaction and annotation for data visualization. Chi [37] developed the Data State Model by breaking the visualization pipeline into four distinct Data Stages with the goal of transforming data into a visual state.

The visualization pipeline offers an easy-to-understand dataflow model that transforms data from its raw form into a data structure that supports visualization [37]. The architecture combines six different independent phases into a data transformation pipeline. During the first half of the pipeline, the data undergoes various transformation until it is in a structured that compliments visualization. The second half of the pipeline centers around user interactions and exploration. The final phase produces a visual display from which the user can analyze the data and refine their exploration, which leads to the exploration sub-process cycle.

## 2.2 Abstraction

Infovis provides a robust infrastructure to display data. However, examining data collections becomes progressively more challenging as the data volume grows. Exploring data on a display screen with a few hundred items might be easy, but when the number of items increases to thousands, millions or larger, it may be challenging to establish an overall understanding of the dataset or find items of interest. Without proper filtering, data transformation, or data reduction, large amounts of data can lead to overcrowded displays. Abstraction helps reduce large volumes of data to a level that has the right amount of information to represent the message of the underlying dataset [38].

Ayers [12] argues that *'second generation'* computer forensics tool use abstraction to improve human comprehension and productivity by presenting data at higher-level of abstraction. Shneiderman [34] claims that just like in other fields, abstraction can be used in digital forensics to find patterns, groupings and gaps among digital evidence. Turnbull and Randhawa [39] used higher-level abstraction as one of the core elements in their system design. The system design used abstraction to hide the unnecessary technical details away from the user but still maintain a connection to the original data. Similar to this research, the abstraction techniques reduced the dataset and focused on displaying high-level system events into an intuitive graphical display.

### 2.2.1 Temporal Event Abstraction.

There are a number of previous works with a focus on using temporal information for system event detection. For instance, the Cyber Forensics Time Lab (CFTL) created by Olsson and Boldt [6] extracts timestamps from a wide array of files while maintaining specific metadata about the source events. For future

work, the authors suggested automating the search for predetermined patterns of suspicious system events. Hargreaves and Patterson [26] created the Python Digital Forensics Timeline (PyDFT) that combines multiple 'low-level' events into 'high-level', human-understandable events.

The event detection abstraction for this research uses event sequencing [40] to order system events based on timestamps. Event sequencing takes advantage of knowing that system events produce a finite set of timestamps during execution lifecycle [40]. For example, when a Microsoft Windows operating system program is executed, the executable creates various system file, log and registry entries. It is highly likely three separate traces from those sources with the same timestamps have some type of relationship. The expert rules exploit this behavior to identify high-level system events. This research experimented with five abstractions based on the contents of a test image of a hard disk drive.

Temporal event abstraction follows those temporal breadcrumbs left behind by system execution lifecycle. Following those temporal breadcrumbs back to sources system event to determine the sequence of events and evidentiary support to their execution [40].

Identifying the source of an event using discrete low-level events as demonstrated in [26] exploits the fact that there *'are distinct event starting times and there are a finite number of events that can occur at the same time'* [40]. The output results is a reduced dataset. This reduction is critical in overcoming the constraints of a screen display while presenting an overview of the entire dataset and still maintain connections to trace(s).

## 2.3 Temporal Analysis

Temporal analysis leverages the unique temporal property of data [36] to direct an investigation and reconstruct past events. When enhanced with multiple temporal data sources, temporal analysis can help rule out specific hypotheses, identify evidence that needs further processing, and detect other potentially critical evidence with confidence [23].

Inglot et al.[23] describes two temporal analysis methodologies. The first option, Traditional, used by a majority of tools, only extracts file system timestamps or Modified, Accessed and Changed (MAC) timestamps. This method is known to be unreliable because of skipping key artifacts (e.g., log files, registry entries, recycle bin entries). Additionally, there are well-known techniques and software for changing file system timestamps [23]. The more reliable and preferred method uses specialized software to automate the extraction of timestamps from multiple sources such as log entries, registry key entries and recycle bin to create a Super-Timeline.

The disadvantages of a Super-Timeline is that it produces a vast number of artifacts that causes information overload [23], [41]. Additionally, only a small percentage of digital forensics tools can process the heterogeneous data produced by applications that generate a Super-Timeline[32].

## 2.4 Digital Forensics Timelines

One of the most frequently used temporal analysis techniques is generating a timeline [42]. Timelines are an essential part of any investigation, including a digital forensics investigation. Not knowing the chronological order of system events in a digital forensics examination makes event reconstruction a complicated task.

Timestamps provide vital temporal information for reconstructing events. Using the timestamp data on a graphical timeline provides an overview of system events and makes it easy to detect data profiles, investigative gaps and other potential evidence sources [42]. In a post-graduate study, Prasad et al., [43] instituted a timeline as the first step to help investigators establish a complete understanding of a crime.

As valuable as temporal information is to an investigation, the related work mentions that industry standard tools lack the capability to exploit this information [12] effectively. Recently developed computer forensics timeline tools and research development mostly focus on either evidence collection or presentation [12] with not much attention given to analysis and event reconstruction. Consequently, the analysis phase still consists of mostly manual processes as automation advancements are focused on the other parts of the digital forensics process,leaving timeline analysis techniques outdated. For example, Log2timeline [41], developed by Kristin Gudjonsson, is considered the cornerstone of timeline generation (text-based) in the digital forensics community. It can extract an extensive array of temporal metadata from different sources (file system, recycle bin, registry files, link, etc.) but does not have infovis capabilities. In their report, of the 16 tools evaluated, Carbone et al., found only one graph-based tool, Aftertime, that effectively integrated timeline generation with visualization.

While timelines are valuable in most investigative fields, they are not utilized in every case. Investigators must decide if there are any benefits to using one for their particular case. Nevertheless, their usage seems to be ubiquitous among industry standard tools [23].

The next section presents the two types of timelines. The first section presents the two popular digital forensics tools and their use of text-based displays. The

next section presents the rarely found graph-based displays for temporal data.

### 2.4.1 Text-based Timeline.

Industry standard digital forensics toolkits like EnCase, FTK and SleuthKit have extensive capabilities to conduct a detailed analysis of digital artifacts, but mostly use text-based displays that quickly become crowded [21]. Previous work trend showed that text-based displays are the preferred method for tool developers. The review also showed that using a text-based timeline for temporal analysis is labor intensive and overwhelming due to the number of entries in the display [16], [44], [25].

The *'information overload'* effect can be seen in the text-based display in Figure 4. Figure 4 is an extract from a log2timeline CSV output. An overcrowded display makes an already challenging task much more difficult.

| | datetime | timestamp_desc | source | source_long | message |
|---|---|---|---|---|---|
| 2 | 1970-01-01 00:00:00.000 | Expiration Time | WEBHIST | MSIE Cache File URL record | Location: Visited: Mr. Evil@about:Home Number of hits: 2 Ca msiecf |
| 3 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet001\Control\Sessio winreg/appcompatc |
| 4 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet001\Control\Sessio winreg/appcompatc |
| 5 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc |
| 6 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc |
| 7 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc |
| 8 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc |
| 9 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc |
| 10 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc |
| 11 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc |
| 12 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc |
| 13 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc |
| 14 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc |
| 15 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet001\Control\Sessio winreg/appcompatc |
| 16 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc |
| 17 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet001\Control\Sessio winreg/appcompatc |
| 18 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc |
| 19 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet001\Control\Sessio winreg/appcompatc |
| 20 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc |
| 21 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet001\Control\Sessio winreg/appcompatc |
| 22 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc |
| 23 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet001\Control\Sessio winreg/appcompatc |
| 24 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc |
| 25 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet001\Control\Sessio winreg/appcompatc |
| 26 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc |
| 27 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet001\Control\Sessio winreg/appcompatc |
| 28 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc |
| 29 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet001\Control\Sessio winreg/appcompatc |
| 30 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc |
| 31 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet001\Control\Sessio winreg/appcompatc |
| 32 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc |
| 33 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet001\Control\Sessio winreg/appcompatc |
| 34 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc |
| 35 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet001\Control\Sessio winreg/appcompatc |
| 36 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc |
| 37 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet001\Control\Sessio winreg/appcompatc |
| 38 | 1970-01-01 00:00:00.000 | File Last Modification Time | REG | AppCompatCache Registry Entry | [HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc |

**Figure 4. Log2timeline CSV output.**

Another example is the Encase text-based in Figure 5.

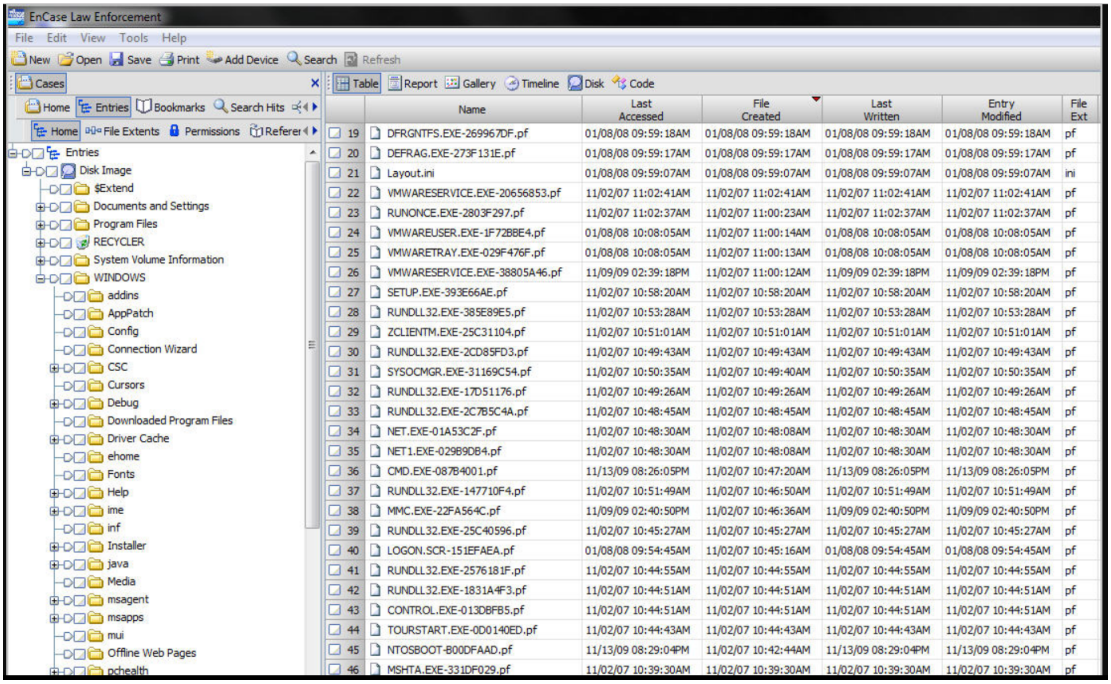



**Figure 5. Encase Text-based Timeline Display [4].**

Shneiderman et al. [16] noted conducting exploration via frequent scrolling within a display that only shows parts of the data hinders data analysis. Additionally, there are separate displays for different file types with no apparatus to establish connections between files other than by manual processes [22].

The authors in [44] and [9] demonstrated that graph-based visualization integrated into digital forensics tools significantly reduce the analyst's cognitive workload when compared to the workload required when using a textual-based visualization. Carbone et al. [32] found only one tool of the 16 reviewed used a graph-based interface.

### 2.4.2 Graph-based Timeline Analysis Studies.

Human experiment results showed that examiners are more efficient and effective using a graph-based timeline versus a text-based timeline [44], [45]. Carry [46] concluded that graph-based visualization reduced the complexity of a dataset

by displaying an overview of the dataset and highlighted significant system events and relationships across the collection. Gujónsson [41] acknowledges examiners are always contending with limited resources while struggling to satisfy increased demand for analysis results. The next section discusses two popular graph-based timeline tools: treemap and histogram.

#### 2.4.2.1 Treemap.

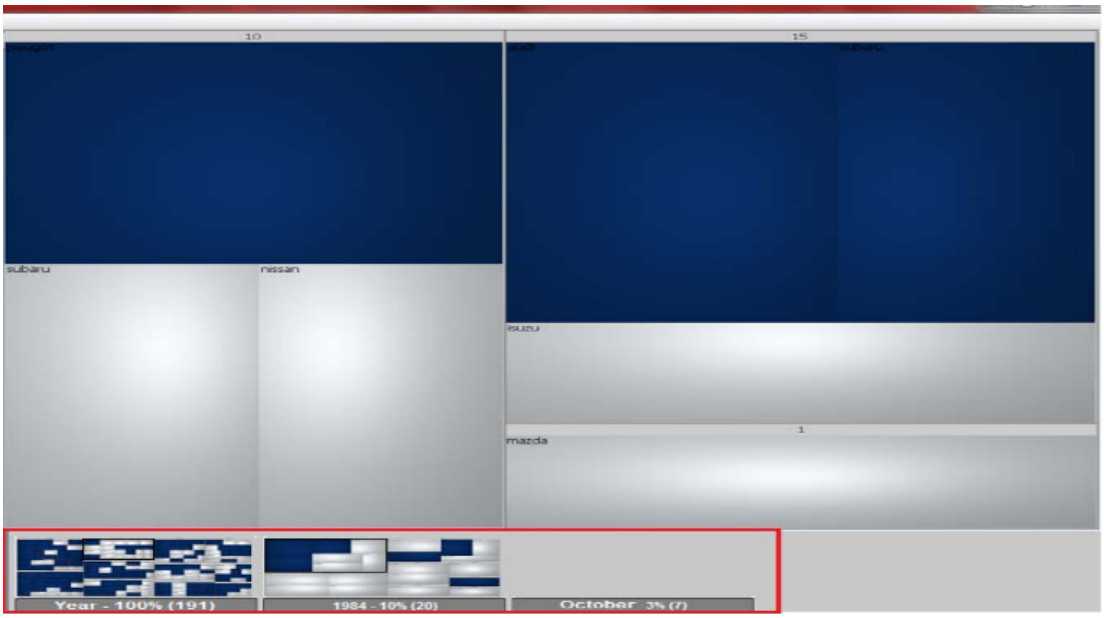



**Figure 6. Temporal Timetree Display [5].**

Carvalho et al. [5] used a treemap visualization to display temporal data. The map provides a hierarchical view that starts at years on top and drills down to days. The design exploits the hierarchical and grouping characteristics of the treemap visualization technique. The infovis technique filled the available screen space and visually organized data into hierarchical data groups using rectangular shapes to abstract the proportional of the frequency of data in the database. The authors choose treemap design to leverage the temporal property of data to help examiners eliminate the 'information overload' challenges. The advantage

of the treemap makes it easy to spot patterns. However, it could be challenging to see the relationship between events with the views stack on top of each other. Additionally, although treemap works great for displaying hierarchical data, as the size of the database increases data the screen space becomes overcrowding which makes analysis difficult. Furthermore, the more frequently occurring data takes up a significant portion of the space leaving little space of the significantly smaller scale [5].

#### 2.4.2.2 Histogram.

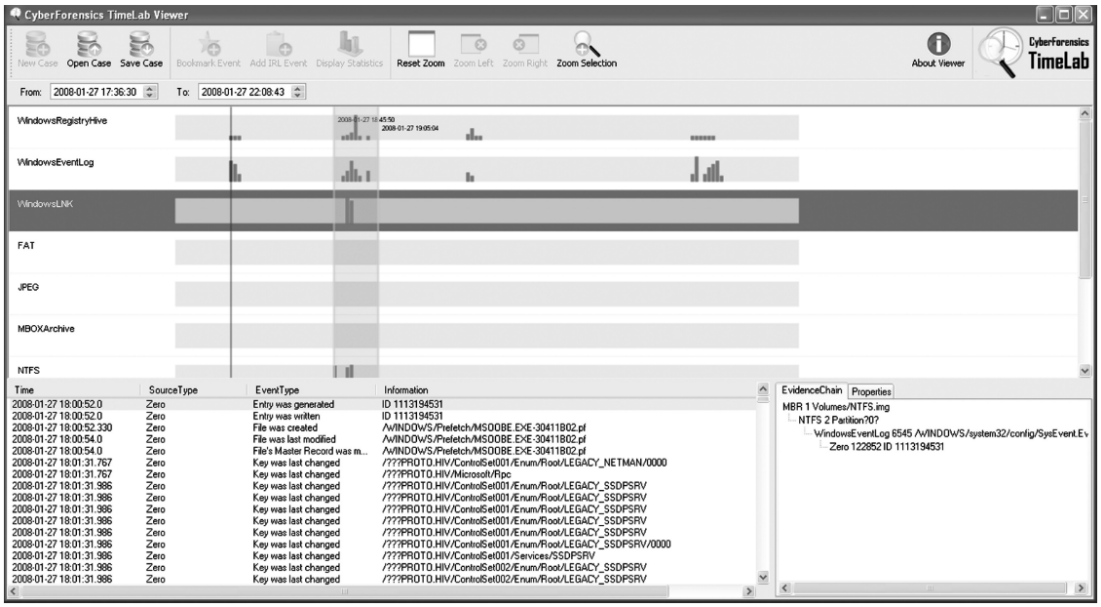



**Figure 7. Computer Forensics TimeLab (CFTL) Display [6].**

One system that does combines a database, GUI and a graphical timeline is Computer Forensics TimeLab (CFTL). Olsson et al. [6] created a prototype digital forensics tool called CyberForensics TimeLab (CFTL). The prototype has two separate parts: a Scanner and the Viewer. The scanner examines an evidence image file recursively then identifies and stores all the timestamps of the dataset. Furthermore, the scanner outputs results using an XML format that is read by

the viewer. The viewer then indexes the entries from the results before rendering the graphical timeline.

The GUI displays the timeline as a histogram for temporal analysis. The bars of the histogram is a representation of the amount of evidence at specific time intervals. There are two notable problems with CFTL. Firstly, the more lightweight JSON (Javascript Object Notation) has become a popular alternative to the verbose XML (extensible markup language) [47]. Furthermore, XML uses alot of unnecessary words in the code which makes it bulky and slow when processing large files [47]. Secondly, CFTL has not had an update since 2012.

## 2.5 Human-Computer Interaction (HCI) and Usability Testing

The previous section discussed how graphical displays can outperform text-displays. This section discusses how to use Human-Computer Interaction (HCI) to choose the graphical display.

The field of Human-Computer Interaction (HCI) commits to understanding how people interact with computers by evaluation of a interactive system design and implementation [48]. HCI is a multi-domain discipline that incorporates techniques from other sciences like psychology, ergonomics, and cognitive sciences to improve human's interaction when interfacing with computers [48]. Modern, advanced GUI's are thriving as a result of implementing HCI best practices to cater to the user's cognitive needs and abilities [48]. By implementing appealing graphics and enabling users to drive innovation, the system design then centers around the human's visual perception. To accomplish this goal HCI conducts usability testing.

The next section presents an overview of one of the critical components of HCI; the usability study. The usability testing as part of this experiment followed

the usability study methods discussed in the next section.

### 2.5.1 Usability Testing.

Lam et al. [49] developed a guide for infovis researchers to find the most suitable evaluation method to achieve their research goals. The authors systematically reviewed 800 papers and detailed infovis evaluations into seven scenarios. In the article, evaluating user experience (UE) is endorsed as the favored measurement instrument to empirically validate the effectiveness, efficiency, intuitiveness and appeal of a visualization capability to support visual analysis and accelerate digital evidence detection. Furthermore, the authors noted that compared to other mainstream evaluation methods, UE combined the collection of both quantifiable metrics such as task completion time and task accuracy and qualitative metrics in the form of personal feedback via participant opinions on the quality of the data analysis experience.

Evaluation in UE *'seeks to understand how people react to a visualization* '[49] which is in tandem with the goal of this research. Shneiderman [36] argues that by recording results from usability testing sessions through observations, interviews, surveys and logging an application's efficiency can be determined. In [50] the authors conducted a user study consisting of a controlled experiment and survey. The study evaluated the effectiveness of the visualization in supporting visual data analysis by conducting usability testing. Furthermore, the authors used open-ended questions to find out to what degree the visualization supported independent hypothesis generation. In [51] the study evaluated five data visualization tools and graded the tools based on their ability to generate 'insights' from the data. More importantly, the authors developed a new testing procedure and a set of measures that combine elements of a controlled experiment and usabil-

ity testing methods. Finally, the experiment design in [3] evaluated a prototype application that generated an interactive visualization of provenance data using a spatiotemporal technique. The controlled experiment consisted of a user evaluation to explore how different history mechanisms impacted problem-solving in visualization environments.

**Summary**

In summary, the literature review of current digital forensics tactics, tools and techniques identified a gap. Several research studies called for an aggressive effort to increasing infovis practices in digital forensics tools. Other researchers summoned the community to develop innovative improvements to digital forensics tools with infovis integration. In two reports, the authors specifically declared no single tool supports the digital forensics analysis by providing access to low-level details from a case-wide, high-level vantage point, in one display using forensically sound procedures [22], [32].

This research contributes to the body of knowledge by demonstrating the effectiveness, efficiency, ease of use and usefulness of a graphical information visualization timeline in supporting digital forensics analysis and digital evidence detection.

# III. Temporal Analysis Integration Management Application (TAIMA)

This chapter presents the Novel Analysis Integration Management Application (TAIMA). Following is a workflow illustration that describes how a forensics examiner can use TAIMA to investigate a system hard disk drive and gather facts about certain system activities by reconstruction events using a graphical timeline. As an illustrative example, a 65G test system image was analyzed and specific system events were first identified using another tool to establish a baseline. The five high-level system events were: program installation, power events (startup/shutdown), program executions, file download and web history.

This demonstration does not discuss the acquisition and validation stages of the digital forensics process. Those processes are not the focus of this study. Furthermore, in the interest of this discussion, it is presumed that the image and the artifacts contained within were acquired through forensically-sound means.

TAIMA is a multi-layered framework designed to assist examiners during forensics analysis. Built using the GRANDStack (GraphQL, React, Apollo, Neo4j Database) [28], TAIMA integrates information visualization (infovis) techniques and provides an effective way to organize and investigate a digital evidence collection. TAIMA was created for this research to mitigate the challenges of visualizing large volumes of heterogeneous data. The design is a combination of several tools, explicitly re-purposed for this research.

TAIMA extends previous work done Schelkoph [52]. Schelkoph developed Property Graph Event Reconstruction (PGER) to store system event data using Neo4j, a native graph database, as storage. In the future work section Schelkoph stated:

*"The ideal situation would enable a user to simply identify a set of objects*

*or actions within a certain time frame that indicates a high-level event. The standardized interface would then interact with the database and provide the abstraction, requiring no special programming skills."*

TAIMA integrates infovis technologies and provides an effective way to identify suspicious files during a digital forensics analysis investigation. Users have direct control to search the database for artifacts based on the temporal attribute of the evidence. As an extension to [52] TAIMA uses also uses a native labeled property graph storage solution that leverages quick path, index-free node traversals to find high-level abstract system events.

## 3.1 GRANDStack (GraphQL, React, Apollo, Neo4j Database)

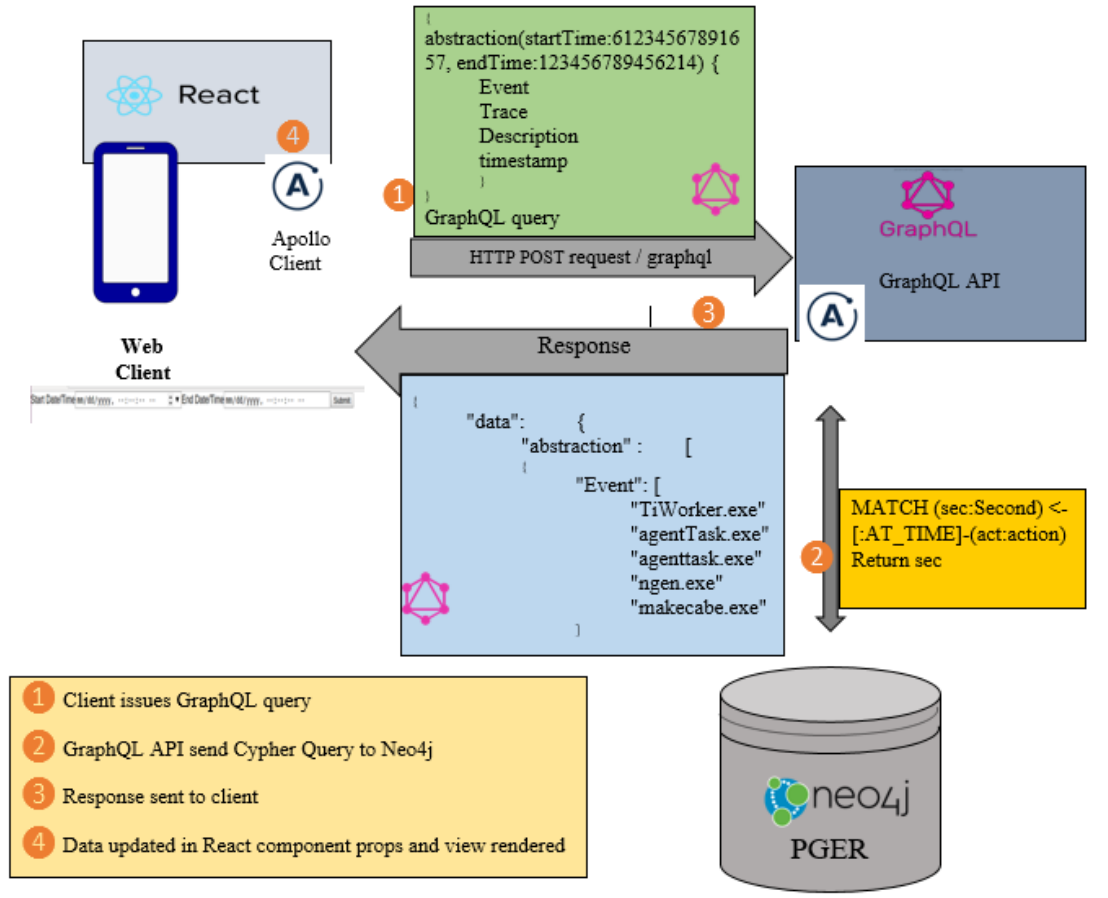



**Figure 8. GRANDStack Architecture [7].**

Figure 8 presents the architecture and state-of-the-art technologies used by TAIMA to visualize system events on the timeline. GRANDStack is an ecosystem of four state-of-art software applications to create full-stack web and mobile grahical user interface (GUI). The integration allows for scalable JavaScript web application backed by a Neo4j database. Furthermore, the integration between GraphQL and Neo4j establishes a robust schema defined database model for fetching data. Using GRANDStack for this research brings the productivity and performance of state-of-art tools to current digital forensics challenges.

The next section describes each of the four application that make up the

GRANDStack and the configuration applied for them to work together seamlessly to provide user-centric exploratory functionalities.

### 3.1.1 GraphQL.

GraphQL is a query language for Application Program Interface (API) technology [53]. Essentially, it establishes an agreement between the front-end and back-end on what type of data can be requested from the database. The response from query forms the application data model [53].

Developed by Facebook, the goal of GraphQL is to translate a client application's data request using an intuitive and flexible framework. It allows developers to specify exactly what data they need [54]. Requesting data using GraphQL consists of two main parts: a schema definition and resolve functions.

### 3.1.2 Apollo Client.

The Apollo Client framework is built to integrate with GraphQL applications to process data fetching and management. Made for both client- and server-side integration, Apollo uses INMemoryCache to store data in the local store in a flattened data structure [55].

To request the data a Cypher query uses the Node.js package to connect with the database and presents the data in a JSON data structure to the React front-end. The front-end stores the results in a local store and makes it available for the visualization rendering.

### 3.1.3 React (JavaScript Library).

The React interface contains two components written in JavaScript: (1) An input field to filter the dataset by time and (2) A visual timeline component.

React is a JavaScript library developed by Facebook that is used for building interactive user interfaces. React can also be used to create mobile applications. Sophisticated React applications usually require additional libraries for specialized state management, routing, and API interaction. Of note, TAIMA uses the vis.js timeline JavaScript library [56]. The front-end interface provides the following five modes of operation: (i) Filter (ii) Zoom (iii) Panning (iv) Details-on-Demand via tooltip

To increase performance and minimize memory usage React maintains an ordered index of all events on the timeline in the locale-store in an array. The React refetch function seamlessly sends a request to the database based on the display need from the GUI. If the view-point-changes the timeline is updated and the data is fetched from the array seamlessly. In the array, each event contains another ordered list array of traces with the event ID as the index; much like a tree with events as the branches and traces as the leaves.

### 3.1.4 Neo4j.

Neo4j is a NoSQL, native graph database opened to the public. It is ACID (Atomicity, Consistency, Isolation, Durability) transactions compliant and uses native property graph modeling [7]. Developers use native graph, like Neo4j for rapid, index-free traversal. Native property graph model stores data using nodes and edges linked by relationships. Additionally, support for index-free graph traversal enables rapid data fetching procedures [7].

### 3.1.5 Rendering the Grahical Timeline.

Rendering TAIMA's graphical timeline is a four-step process. Initially (1), the user is presented with the React user interface frontend to enter a time interval

of interest. After clicking the submit button, the Apollo client sends a GraphQL query to the Neo4j GraphQL (2) service with the timestamp as search parameters. The GraphQL server contains logic on how to query the Neo4j database to search for high-level events based on their temporal attribute. After fetching the data, Neo4j Apollo client (3) sends back the results to the client Apollo service. The React integration for the Apollo Client is configured to store the results of the GraphQL query inside a React component to render the visualization (4).

## 3.2 Data Transformation

Referencing the Visualization Pipeline from Chapter 2, this section describes the six phases of the data transformation process of TAIMA.

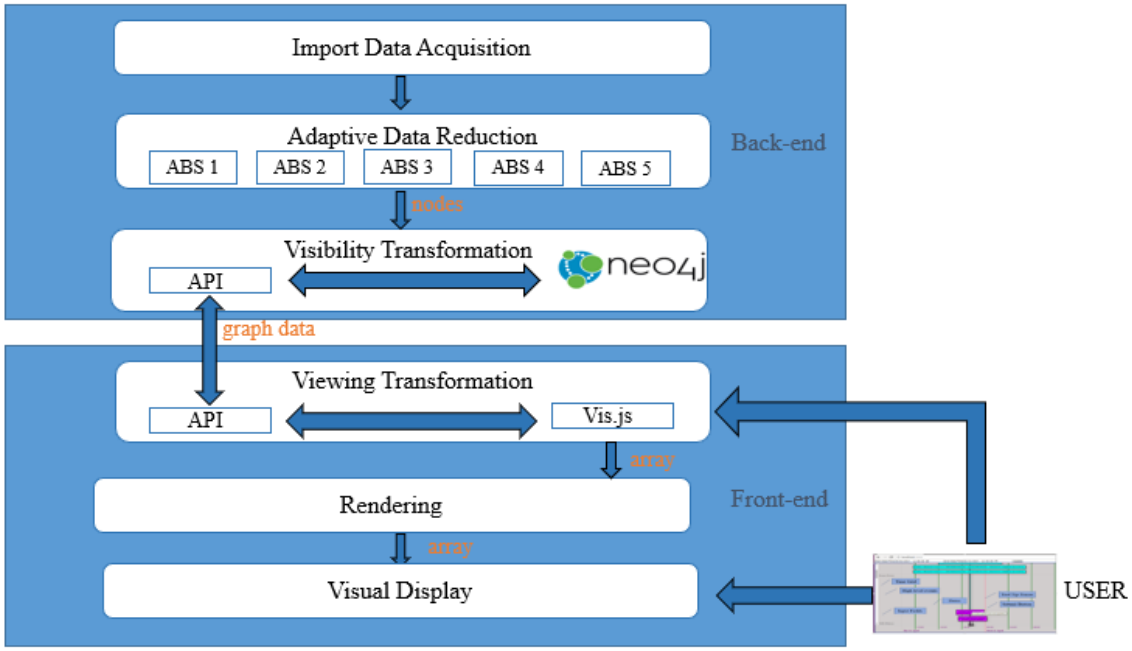



**Figure 9. TAIMA Data Transformation Visualization Pipeline.**

### 3.2.1 Import Data Acquisition.

The first process, Import Data Acquisition, loads the Neo4j database from PGER into the Neo4j databse instance connected to TAIMA.

### 3.2.2 Adaptive Data Reduction.

The second phase, Adaptive Data Reduction, uses abstraction to reduce the size of the subgraphs that includes system events of interest. The goal of this phase is to implement a high-level of abstraction that expresses important information about the particular system events while minimizing unnecessary information [38].

The abstraction technique mines low-level traces based on predetermined expert rules. If the criteria of the rule is satisfied a high-level event node is added to the graph model to represent that system event. Then, new relationships are added to abstraction node to track the traces connected to the high-level events. Aggregating low-level events and linking them to higher-level events not only reduced the amount of data presented to the analyst but also increase the efficiency of the application (uses less memory).

### 3.2.3 Visibility Transformation.

The third phase, Visibility Transformation, converts the abstraction nodes into the visual data elements upon request from the front-end. To maintain the integrity of the original data source GraphQL is used to request the data from the database.

#### 3.2.3.1 GraphQL.

GraphQL provides a comprehensive description of the datastore in the Neo4j database. The advantage of GraphQL is the ability to define a schema that describes exactly what data to request and nothing more [54].

**Schema**

The schema defines how data is fetched from the database [54]. It provides an outline of the available data type in the database [55]. Additionally, it defines the

specifications for the database API. Figure. 10 is a part of the schema used by TAIMA .

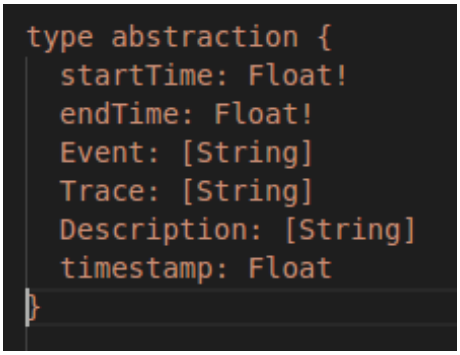



**Figure 10. TAIMA GraphQL Schema.**

Figure 10 shows the type definition of abstraction, the main entry point to the database. The definition list the data will be requested.

**Resolver Functions**

Figure 11 shows TAIMA uses Neo4j GraphQL resolver function. The Neo4j GraphQL resolver functions contains application-specific functions that defines how to fetch data based on a one to one mapping with the fields in the schema [54]. Moreover, it is responsible for translating the GraphQL queries into a Cypher query. After executing the query, the results of the query are returned to React as an array of objects.

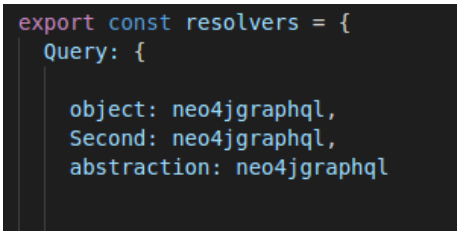



**Figure 11. TAIMA's Resolver Functions.**

Figure 12 shows the query type abstraction that is executed to request data from the Neo4j database. The abstraction query takes two arguments: startTime

and endTime. The exclamation point indicates that both are required in the query request. By specifying the fields in the request, the query traverses the Neo4j database to find and return the values for those fields. The abstraction query returns an array of abstraction objects.

```
abstraction(startTime: $startTime,  endTime:$endTime)
  {
    Event
    Trace
    Description
    timestamp
    group
    className

  }
```

**Figure 12. Query Type.**

### 3.2.4 Viewing Transformation.

Figure 13 shows the mapping of the abstraction objects key/value pairs to vis.js parameters. At this stage the framework and libraries have the data structure they need to render the infovis.

```
//Viewing Transformation:Mapping the raw data to the timeline format.
const newArray = data.abstraction.map((abstraction, index) => ({
            id: index + 1,
            content: abstraction.Event.join(",<br>"),
            start: abstraction.timestamp,
            title: abstraction.Trace,
            end: null,
            group: abstraction.group,
            className: abstraction.className
    }) )
```

**Figure 13. Visual Mapping Transformation.**

### 3.2.5 Rendering.

The fifth stage, Rendering, is a coordinated effort between React and the vis.js timeline JavaScript library [56]. The vis.js library verifies the data meets the visual displays structure. React then loads the timeline as a React component

to display on the screen.

### 3.2.6 Visual Display.

The final stage, visual display, presents the graphical user interface (GUI) to user. The GUI first presented an overview of all the events included in the time interval. Using panning and brushing the user is able to zoom and filter events on timeline. And lastly, to fulfill the Mantra mentality, details on demand is provided via tooltip that displays the traces connected to the high-level event.

## 3.3 The Interface

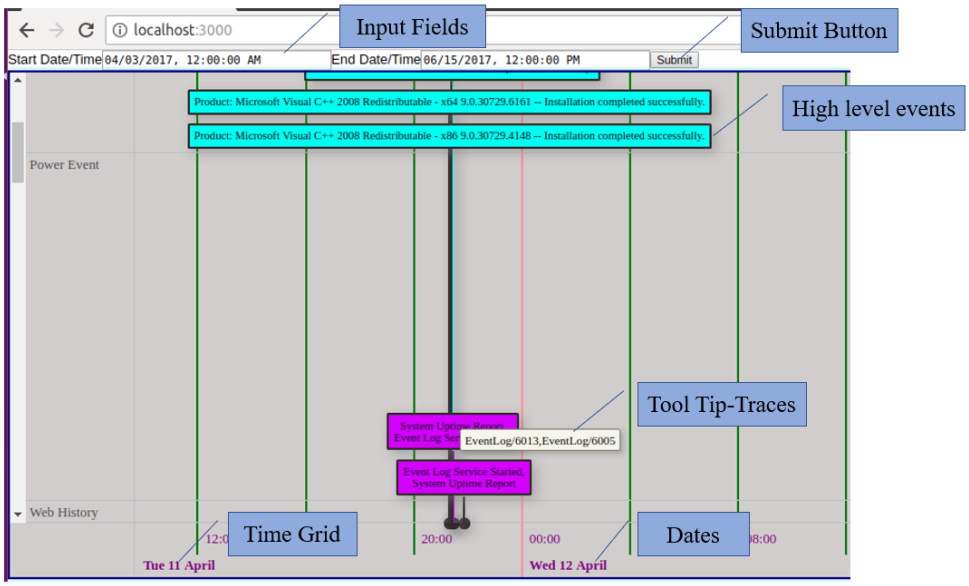



**Figure 14. TAIMA Interface.**

The user interface has two input fields from which to select a time interval. After clicking the submit button a timeline populated with system events is loaded. The timeline is an interactive visualization component that transforms discrete timestamp data into a graph form. All the high-level systems events are listed from left to right and order chronologically based on their temporal property.

Enriched abstraction nodes provide details-on-demand using the tooltip functionality. Hovering the mouse over an event tile displays the event trace(s).

On the timeline, an event is represented as a title and can include just one trace item or several as an abstract event. The number of events in a tile is determined by how many events have that same timestamp. As part of an event the timestamp is stored as a UNIX timestamp (or Epoch), is unique in the database and represents the time of an event.

Integrated with basic visualization controls, the user can click, drag or zoom in/out on the x-axis. The time scale on the horizontal axis adjusts from milliseconds to years automatically based on the user's desired field-of-view. This presents the examiner with precisely what they are interested in. By exploring the timeline, the examiner can get a general understanding of the activities that occurred on the system during a specific time frame; all in one view.

### 3.3.1 Sheiderman Requirements.

Shneiderman formalized seven tasks to evaluate the effectiveness of a graphical user interface for infovis, which helped model the design of TAIMA [36]. Table 2, shows how TAIMA fulfills six out of the seven tasks. In this current version, history is currently not maintained. However, in future versions keeping track of the user's interactions could be added along with buttons to undo previous actions.

**Table 1. TAIMA information visualization meeting Shneiderman Task Requirements.**

| Task | Description |
|---|---|
| Overview | The timeline populated with the results from the query |
| Zoom | Changes smoothly from an overview to a close-up or vice versa |
| Filter | Removes unwanted data from point-of-view on timeline |
| Details-on-Demand | Clicking on a tile displays a tooltip containing the full file path of the object |
| Relate | View temporal relationship on the timeline |
| History | *Not implemented |
| Extract | Print screen |

## 3.4 The Data Model

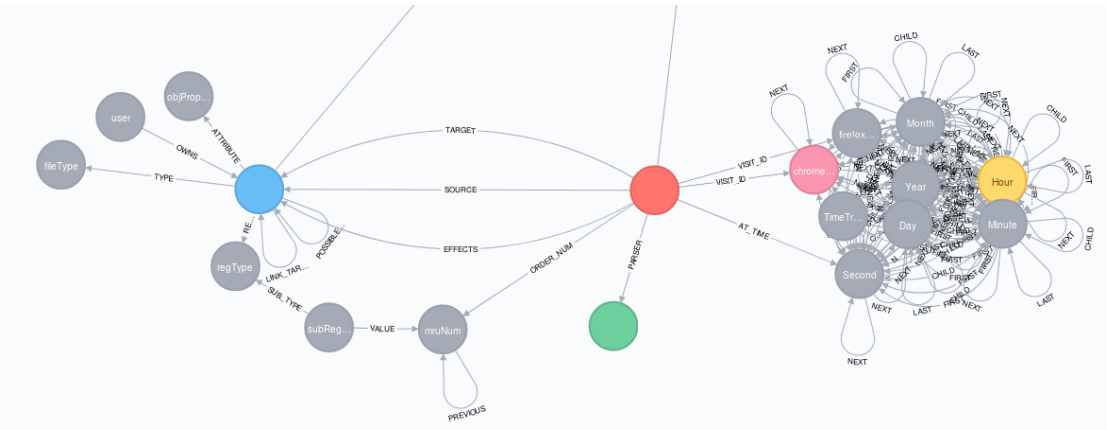



**Figure 15. PGER Data Model.**

Figure 15 is an schema representation of the data model. On the right is a timetree which is connected to the rest of the graph via the red node. The red node represents the action nodes and is one of the fundamental elements that make up

the graph. The action node is a graphical representation of a system activity that altered or changed system files such as, Keywords Searched or changes to MAC times [52]. The action nodes affects a digital artifact; represented as the object node [52]. The digital object stores the name of system artifacts such as, a URL, registry key, or file path [52]. The action and object nodes have three connections to each other: TARGET, SOURCE and EFFECTS relationship. The direction of the relationships is from an action (red) to object node (blue). The relationships are leveraged to create the enriched high-level abstraction events that populates the TAIMA's graphical timeline in addition to keeping a record of the low-level events that are attributed to the high-level events.

The abstraction queries includes logic that mines the datastore and adds abstraction nodes after importing the dataset. Building the queries required an analysis of the database graph model. The analysis provided insight into the structure of the data model and the most efficient traversal paths.

The data model consists of the following nodes: action (red node), object (blue node), parser (green node), and a timetree. The nodes have the following significant relationships:

1. action TARGET - SOURCE - EFFECTS object
2. action VISIT_ID chrome
3. action VISIT_ID Firefox
4. action PARSER parser

The object is the artifact affected by the action process and points to a URL, registry key, or file path [52]. Figure 16, lists the node and relationship count in the Neo4j database after pre-processing by PGER.

| Store Sizes | |
|---|---|
| Array Store | 90.55 MiB |
| Logical Log | 106.69 MiB |
| Node Store | 15.61 MiB |
| Property Store | 66.85 MiB |
| Relationship Store | 87.47 MiB |
| String Store | 71.72 MiB |
| Total Store Size | 712.07 MiB |

| ID Allocation | |
|---|---|
| Node ID | 1080246 |
| Property ID | 1709314 |
| Relationship ID | 2667340 |
| Relationship Type ID | 29 |

**Figure 16. PGER Data Model Statistics.**

## 3.5 Temporal Event Abstraction

The analysis revealed the datastore contained over one million nodes. Data reduction techniques provided a way to compress specific slices of the entire dataset. The goal of data reduction procedures is to abstract a subsection of data. While reducing the dataset, reduction techniques must also maintain the data parameters and attributes of the original data. To distill the raw data into more appropriate representations involves data filtering and enrichment. Data is filtered to extract relevant information, while data is enriched with higher level information that supports a given task

The dataset is mined for high-level systems and sorted by time. This allows the examiner to choose a time window of interest. Based on previous research abstracting out system events using the temporal property of data should make finding evidence faster [6].

The abstraction queries creates high-level events from low-level events based on predetermined rules. Each abstraction query has logic that describes a criterion that identifies the low-level events that should be present if certain high-level event

occurred. The temporal abstraction Cypher query searches the entire database for low-level events that match a predetermined logic, within a specified period of time interval [57]. If found the query creates an abstract node of the high-level event and establishes a connection to the low-level events. Once the links are established the provenance of the high-level events are preserved. The following is an example of one of the five Cypher queries. The query creates the Program Installation abstraction:

(Program Installation)

- Event ID: 1033 Records the end result of a program installation. Status code 0 means the installation was successful.
- Event ID: 1042 Notification of the installation process completion.
- Event ID: 11707 Successful installation

Figure 17 shows the Program Installation Cypher Query that abstracts an installation attempt.

```
1     MATCH (:parser {parserName: "eventLog"}) <-[:PARSER]-
      (act:action) -[:EFFECTS]->(event:object)
2     MATCH (act)-[:AT_TIME]->(sec:Second)
3     MATCH p = (sec)-[:NEXT *10]->()
4     WITH p, event
5     UNWIND nodes(p) AS secNodes
6     MATCH (secNodes) <-[:AT_TIME]-(act:action) --(obj2:object)
7     WHERE obj2.filename IN ["MsiInstaller/11707",
      "MsiInstaller/1042", "MsiInstaller/1033"]
8     WITH act.timestamp as timestamp, COLLECT(DISTINCT
      act.message) as messages, COLLECT(DISTINCT obj2.filename)
      as filenames, COLLECT(DISTINCT act) as acts
9     CREATE (a:Absatraction{Event: 'Program Installation',
      Trigger:filenames, Description:messages,
      timestamp:timestamp})
10    FOREACH (act in acts | MERGE (act)-
      [:LVL1_ABSTRACTION_LINK]->(a))
      FOREACH (set in obj2s | MERGE (set)-
      [:LVL1_ABSTRACTION_LINK]->(a))
```

**Figure 17. Program Installation Cypher Query.**

Lines 1-5 search a time window for object and action nodes with relationships to the eventLog parser. Line 6 searches those actions from line 1-5 that are linked to objects nodes labeled as: MsiInstaller/11707, MsiInstaller/1042, MsiInstaller/1033. Lines 8 then COLLECTs (aggregate the nodes based on time) the values into a list of DISTINCT items then returns all nodes linked to each unique time. After creating the abstraction nodes in line 9, line 10 links the action and object nodes responsible for program installation. In Figure 17a the grey node is the abstraction node and is connected to four low-level nodes. Figure 17b is the enrichment metadata stored by the abstraction node.


(a) Program Installation Abstraction node (grey)


```
{
  "Event": "Program Installation",
  "Description": [
    "Windows Installer installed the product. Product Name: Microsoft Lync MUI (English) 2013. Product Version: 15.0.4569.1506. Product Language: 1033. Manufacturer: Microsoft Corporation. Installation success or error status: 0. ",
    "Product: Microsoft Lync MUI (English) 2013 -- Installation operation completed successfully. "
  ],
  "Trigger": [
    "MsiInstaller/1033",
    "MsiInstaller/11707"
  ],
  "timestamp": 1491245982000
}
```

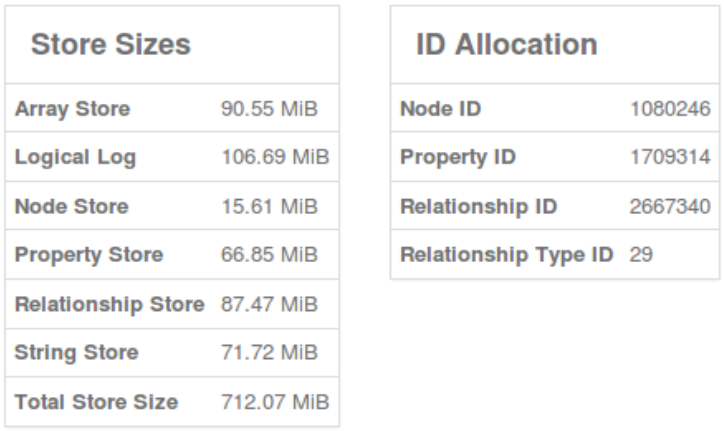

(b) Enrichment Quadruple Information

**Figure 18. Program Installation Abstraction node and Enrichment Information Note: The abstraction nodes provides enriched information by including the quadruple metadata from the LVL1_ABSTRACTION_LINK relationship as: Event, Description, Trigger(Trace), timestamp.**

## 3.6 TAIMA Workflow

This section illustrates how TAIMA processes work together to generate an effective and efficient visualization for examiners conducting timeline analysis. The demonstration will also show how the examiner can gain situation awareness of system events using an intuitive graphical user interface.

As an extension of [52] TAIMA assumes the underlying data is a graph database. PGER extracts system artifacts from various sources from a image (e.g., MAC table, logs, registry, and much more) and converts those artifacts into different subgraphs and store them in a Neo4j database [52]. Figure 19 shows the size of database produced by PGER is 712.07 MiB.



| Store Sizes | |
|---|---|
| Array Store | 90.55 MiB |
| Logical Log | 106.69 MiB |
| Node Store | 15.61 MiB |
| Property Store | 66.85 MiB |
| Relationship Store | 87.47 MiB |
| String Store | 71.72 MiB |
| Total Store Size | 712.07 MiB |

| ID Allocation | |
|---|---|
| Node ID | 1080246 |
| Property ID | 1709314 |
| Relationship ID | 2667340 |
| Relationship Type ID | 29 |

**Figure 19. Size of Neo4j graph store for the 65GB test image.**

The first process of TAIMA is to create abstraction nodes. For this workflow the following five high-level abstraction nodes were created: (program installation, power events (startup/shutdown), program executions, file download and web history). Figure 20 shows the data model after applying the five abstractions (purple node). The purple node is a abstract representation of the five abstraction nodes. The LVL1_ABSTRACTION_LINK relationship links them to the other

nodes in the data model with the action and object nodes as entry points. The contents of the abstraction nodes include enrichment information gleaned from action/object relationships.

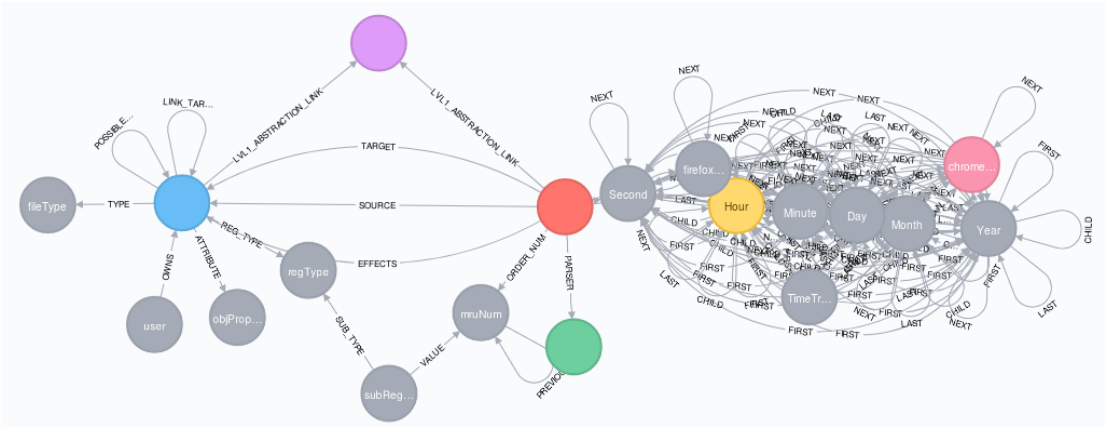



**Figure 20. Test Image Data Model After Abstractions.**

The total node count of the entire database was 1,069,252.

```
$ MATCH (n) RETURN count(n)
```

Table

| count(n) |
| --- |
| 1069252 |

## 3.7 The Interface: Input Fields

Upon loading, TAIMA presents the examiner with two input fields, start time and end time (Figure 21), to specify a time interval. Before submitting the request the date/time inputs are converted to Epoch time. It is worth mentioning that as a React component the input fields are another element that reduces the items displayed on the screen. Only the events that fall within the time interval are loaded on the timeline.

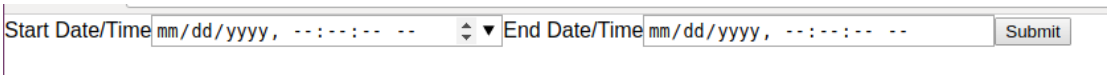



**Figure 21. Interface Input Fields.**

After converting the timestamps a Cypher query is sent as a POST request to the Neo4j database. The query uses the times as parameters to search for high-level events that occurred within the requested time frame. The result is a focused display of only the requested activities.

## 3.8 The Timeline

In addition to creating high-level events, TAIMA renders a graphical timeline. The timeline displays a variety of system activities in sequential order. The high-level system events are represented as colored rectangular tiles to differentiate the five system activities of interest. Upon reviewing the timeline, it is possible to visualize how often certain system activity occurred within a given time period. Additionally, the timeline provides an overview that displays high activity dates as the point-of-view is zoomed out. For example, immediately upon viewing the timeline it was apparent that a majority of the system activities occurred in April and June in 2017 as seen in figure 22 below. The next step further examines the cluster of events in April.

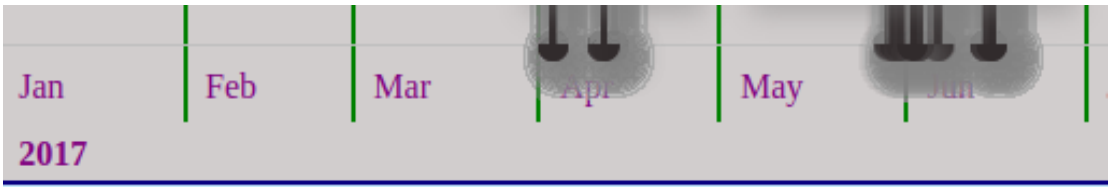



**Figure 22. Year/Month view: Clustering of events on April and June.**

Figure 23 is a zoomed in view of April 2017. The filtered viewed does not show June's activity. The cluster of events in the point-of-view forms around 3 April 2017 and 12 April 2017.

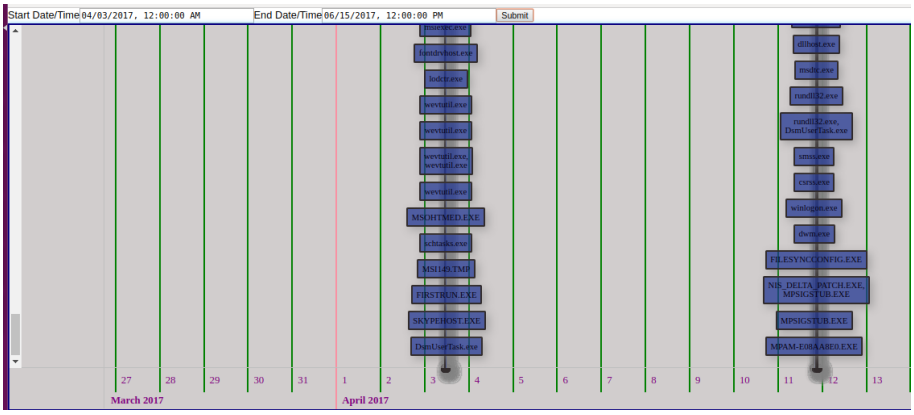



**Figure 23. Month/Day view: Clustering of events on 3 and 12 April 2017.**

Further inspection of 3 April 2017, using the zoom function revealed the following (see Figure 24):

(1) Figure 24 shows various file downloads occurred between 11:40 – 12:10

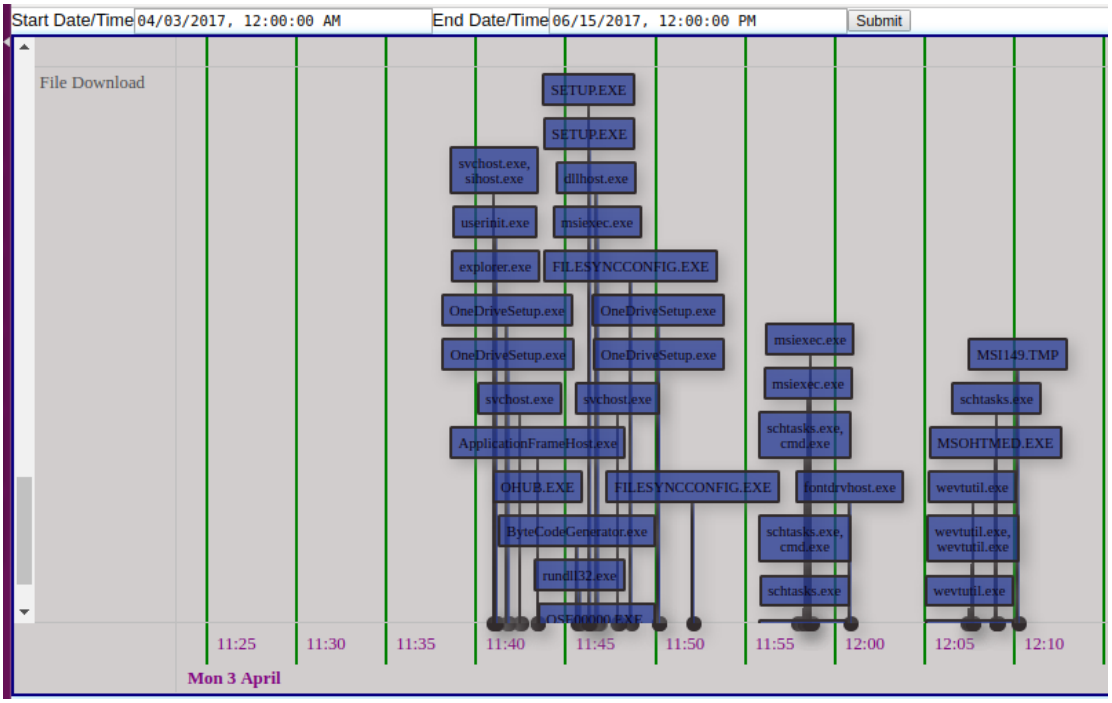



**Figure 24. Day/Hour view of 3 April 2017.**

(2) In Figure 25, on Tuesday, 11 April 2017 around 09:25PM (21:25), VMWare was installed (aqua colored tile). A few minutes later (21:25 and 21:50) the time-

line shows two power event (purple tiles); restarts, probably due to the VMWare install.

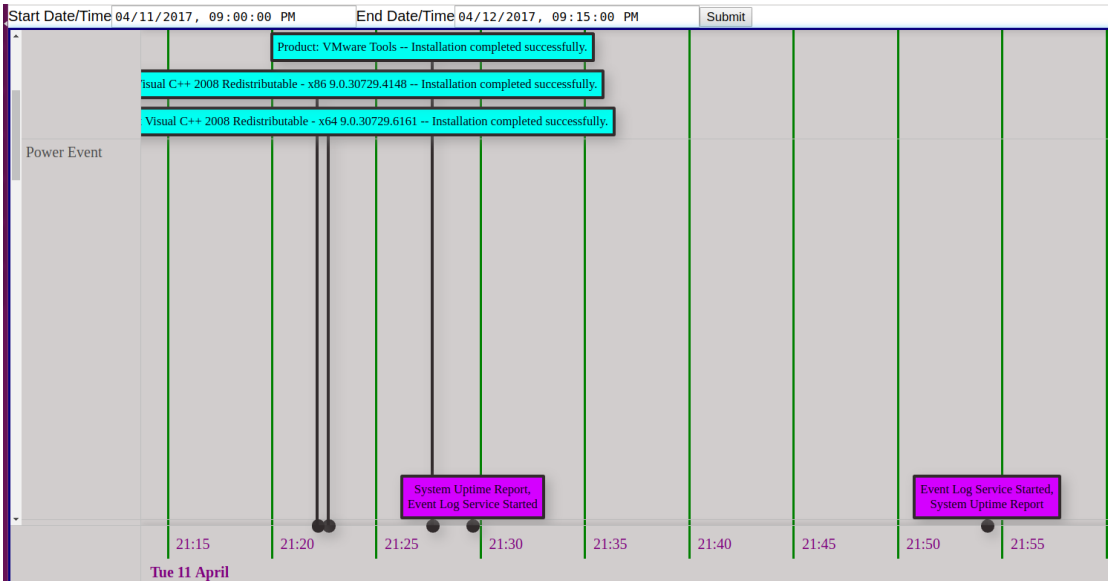



**Figure 25. VMWare Install and Restart.**

## 3.9 Zoom

When zooming in on a specific time period, the x-axis scale changes depending on the magnitude of the zoom as can be seen in Figures 23 and 24. In Figure 23, the scale changed from displaying months and years to days and hours/minutes as seen in Figure 24. Furthermore, zooming also works as another filter. The max zoom position will display an overview of all the events included in the time interval entered via the interface. However, zooming in on a specific time period causes nodes to disappear that do not fit on the point-of-view on the screen. This allows the examiner to focus their attention to only on what is on the screen.

## 3.10 Traces

The enriched abstraction nodes provide not only the timestamp of high-level events but the low-level events attributed and associated to the high-level events.

Hovering the mouse over an event tile displays the source artifact(s) via tooltip. For example, in Figure 26 the tooltip displays the sources triggered by the restart. In this case, the restart that occurred on 11 April around 20:00 hours triggered log entries, EventLog 6013 and EventLog 6005.

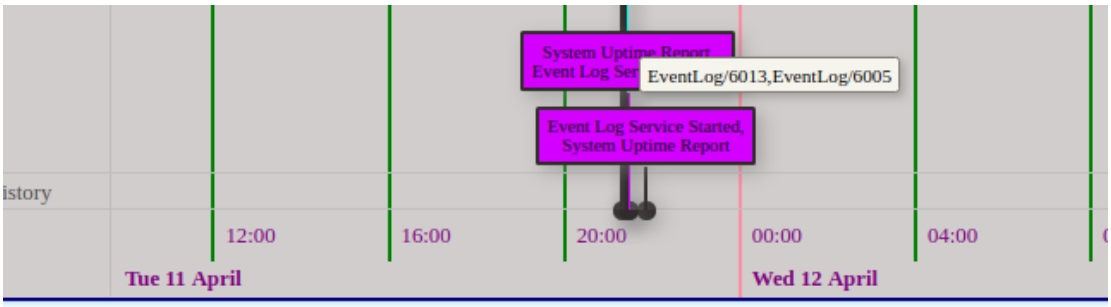



**Figure 26. Tooltip Displaying traces.**

## 3.11 Summary

TAIMA enables examiners to explore system events patterns, temporal proximity and to gain a better understanding of what happened on a system. TAIMA's design focused on data enrichment and visualization best practices to help examiners understand a dataset. The abstraction techniques substantially reduced the dataset while still maintaining the original the integrity of the original data database.

The infovis aims to provide an interactive, user-friendly approach to digital analysis by implementing exploratory and correlation techniques. The visualization is an dynamic, interactive display that presents an overview of all the high-level system events found within a specific time interval. The timeline lists the events in chronological order with the earliest event on left. The examiner can highlight an event of interest by clicking on the tile. The tooltip displays a listing of trace(s) that triggered that particular event. With a little exploring the timeline can provide the examiner an overall understanding of the activities that

occurred on the system, in the order they occurred and which username associated with those activities.

# IV. Research Design/Strategy

The research hypothesis is that an interactive GUI with a graph-based timeline can minimize the challenges of digital forensics analysis. Previous studies show that integrating advanced infovis methods and practices to digital forensics tools reduce investigative timeline and significantly increase accuracy in discovering relevant digital evidence. Infovis modus operandi infuse graphics into an interactive digital environment to support comprehension of complex of data. As a result, relationships and data patterns that might not be identified in a text-based display can be uncovered and recognized easier with infovis applications.

This section discusses the methodology of the research. The section first, discusses the research method and strategy for the development of the user study approach. It is followed by a brief discussion about the decision to use a fictional case disk image and the ethical considerations with using real world disk image. The next section provides support for using between five and eight participants to evaluate the prototype. Then the next section provides details about data collection and the analysis procedures. Finally, this section is concluded with a discussion on the research limitations and strategies to minimize their effect on the results.

## 4.1 Evaluating User Experience (UE)

The motivation for this research is rooted in the need to simplify the analysis phase of digital forensics. The challenges associated with the complexities of the analysis of forensics evidence analysis and temporal event reconstruction attributed to large amounts of heterogeneous data is well documented in [12],[31], [6].

This study followed the Evaluating User Experience (UE) guidlines set forth by Lam et al [58]. UE evaluations include assessments that analyze individual response and attitudes towards a visualization [49]. The UE for this study combines usability testing (UX) and a Post-Study Usability Questionnaire (PSSUQ) based on their collective strengths.

For the usability testing, a test scenario simulated a real-world hacking investigation. The five participants performed a digital forensics analysis investigation using the prototype GUI to identify particular files of interest related to a notional criminal case. To conduct an efficient evaluation of the visualization framework quantitative and qualitative data were collected. Data analysis included participant's task accuracy, time completion (quantitative) and the usability questionnaire (qualitative).

### 4.1.1 Disk Image.

For the usability testing, a test scenario simulated a real-world hacking investigation. The assigned task was designed to simulate the analysis phase of a digital forensics investigation. Furthermore, the assigned task required the participant to use the rendered visualization and inspect the timestamp data to identify suspicious files.

The investigation was simplified due to consideration of participants' time. However, despite the simplification of the test case, it still provided an opportunity to evaluate the prototype's capabilities properly. See Appendix C for details on the details of the task. The fictitious case image was downloaded from the National Institute of Standards and Technology (NIST) Computer Forensic Reference Data Set (CDReDS)[59]. The image is of an abandoned notebook computer that is suspected of being used for hacking purposes. For further details regarding the

nominal case refer to Appendix C.

Participants were provided investigative leads regarding the time when the computer was suspected of being used for hacking. This information could be seen as an advantage for the timeline visualization. However, in real-world cases, examiners are routinely provided with timestamp information about the subject and their activities. As a result, the approximation reflects real-world and thus provides a realistic test case and should not pose a substantial problem for the evaluation of the prototype GUI.

### 4.1.2 Task Description.

The users performed a digital forensics analysis investigation using the prototype GUI to identify particular files of interest related to the notional criminal case. The prototype GUI provides an interactive, graph-based visualization of event-based data with ordinary as well as malicious behavior.

Prior to starting the session the participants were trained on how to use the GUI and given time to explore and ask questions. In consideration of time, the time limit for each session was set to 30 minutes but the participants' were not discouraged from going over that time limit. However, a hard stop limit was set at one hour. Data collections included, journal entries on the total number of hacking software suspected of being used to hack (error rate) and time spent on the interface (performance).

To obtain participants' reactions to the GUI a post-test questionnaire was administered after the task was completed. The survey included demographic questions and questions soliciting feedback from their experience with the GUI. Moreover, the study included both open-ended, free-text answers along with some rating answers (i.e., such as asking for the perceived task difficulty on a scale of

1-7) to gather quantitative and qualitative metrics to conduct statistical analyses.

### 4.1.3 Population Selection.

The recruitment goal was to recruit between 5-8 Department of Defense (DoD) Certified Digital Forensics Examiners (CDFE) from the United States Air Force (USAF) Office of Special Investigations (AFOSI). However, attempts were made to use the maximum number of SMEs available. According to Nielsen, approximately 90% of usability problems are discovered with no more than 5 participants in a usability test [8].

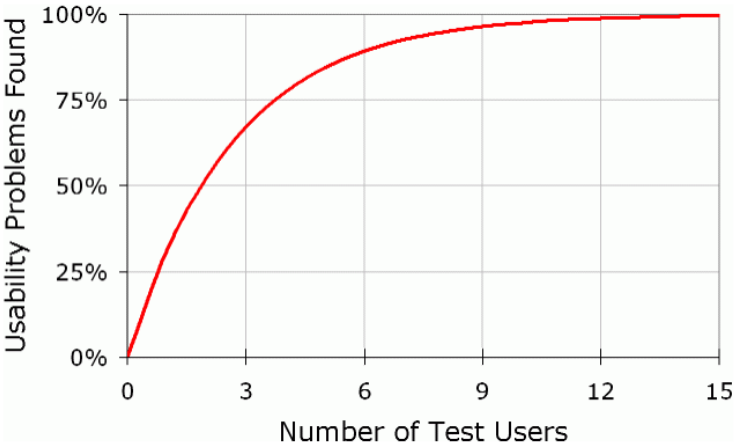



**Figure 27. Diminishing Returns for Usability Testing [8].**

The participants needed to have the required background to understand and complete the task. Minimum qualifications for participants were:

1. Digital forensics experts with experience conducting digital forensics investigations and analysis tools, tactics, techniques and procedures for associating people, locations, things, and events with digital evidence; and
2. members of AFOSI with experience in digital forensics.

The recruitment strategy was to include an equal amount of participants of each sex. Unfortunately, due to the high rate of males in the digital forensics field,

it was not possible to have an equal amount of each sex.

Introduction and hands-on demonstration was conducted with participants for tool familiarization and troubleshooting prior to testing. When the participant indicated, verbally, they were comfortable using the tool the test scenario and objective was issued to start the experiment.

### 4.1.4 Evaluation Technique.

To conduct an efficient evaluation of the visualization framework quantitative and qualitative data were collected. The collection focused on:

- Task performance collected as task completing time
- Error rate (Number of hacking software traces found divided by 6)
- Subjective user satisfaction capture via post-task questionnaire

The focus of the quantitative data collection was on task completion time and accuracy. The analysis report turned in by participants after their investigation described their findings and was reviewed to identify the total number of the hacking software listed in the report (error rate). The focus of the qualitative data collection was from post-task questionnaire and open-ended, free-text answers which supported the analysis of the post-task questionnaire.

After completing the task, participants were asked to complete a Post-Study System Usability Questionnaire (PSSUQ) reflecting on their experience with the GUI. To maintain anonymity, each participants questionnaire was assigned a unique ID number based on the order of their testing (e.g., 1, 2, 3,...). The questions from the usability evaluation were inspired by PSSUQ [60]. Developed in 1992 by IBM, the PSSUQ assesses the participants satisfaction of the GUI in

achieving the task in the usability study. This research modified the original version of the PSSUQ. The original PSSUQ comprised of 19 items. The modified version for this research used only 15 items rated on a seven-point Likert scale (strongly disagree 1 to strongly agree 7).

The PSSUQ consists of an overall satisfaction scale and provides a grade for the overall usability of the GUI based on the participants' responses [60] in three sub-scales: system usefulness (items 1–8); information quality (items 9–15); and interface quality (items 13-15). Higher scores indicate better usability.

Developed in 1932 by Rensis Likert [60], Likert-scale questionnaires prompts for subjective opinion and attitude regarding a visualization tool. Olsson and Boldt [6] used a Likert-type scale to solicit feedback from users of their prototype. The typical Likert-scale is a 5- or 7-point scale and ordinarily used by participants 'to rate the degree to which they agree or disagree with a statement.' The Likert scale is frequently used in GUI usability evaluation as it provides a mechanism to capture subjective assessment of an application's perceived usability [11]. Additionally, literature recommends the use of a Likert scale, particularly when attempting to measure complex concepts—where a single survey item is not likely to adequately convey the intended concept, such as motivation, satisfaction, and confidence [61]. For further details on the questionnaire see Appendix B.

### 4.1.5 Data Analysis Procedure.

Data collection included collecting the results of the participant's task accuracy, time completion (quantitative) and the usability questionnaire (qualitative). Data analysis calculations to evaluate the effectiveness and efficiency of NAIMA includes:

1. Effectiveness: Calculated by averaging the means of the three sub-scores of

the PSSUQ: System Usefulness, information Quality and Interface Quality. Note: higher scores denotes better usability.

2. Accuracy: success rate = (number of task/total number) * 100.
3. Efficiency: the average (mean) time taken to complete the task.

### 4.1.6 Research Limitations.

There are several limitations to this study. For instance, the simulated case data was not actual crime scene data. However, the simulation helped to protect personally identifiable information of actual victims and subjects.

The scope for the study was limited to only finding executables to help reduce stress on participants. However, this limitation makes it hard to ascertain how the product is going to perform over an extended period [25].

Ordinarily, usability laboratories are used to conduct usability tests [58]. The lab environment provides an area that allows the investigators to observe the participants. However, due to limited resources and proximity, the participants conducted the evaluation remotely. To reduced the effect of this particular limitation the participants were asked to turn in an investigative journal that included their thought process used to complete the assigned task in addition to their PSSUQ.

**Summary**

The research design mirrors a User Experience (UE) evaluation to provide empirical evidence that supports the hypothesis that the effectiveness, efficiency, intuitiveness and appeal of the visualization tool's supports visual analysis and accelerate digital evidence detection.

The primary investigative activities were the base procedures for conducting a UE evaluation as described in [58]. To ensure the appropriate sample size was selection this research used the Nielsen Norman Group justification that states,

the majority of usability problems comes from testing no more than five users [8]. Following completion of the analysis task, the participants were asked to complete a post-task self-report questionnaire designed to focus on the usability of the GUI in achieving the task in the usability study [8]. Data analysis included compiling the results of five participants performing the primary digital forensics analysis tasks using the GUI and the visualization.

# V. Results & Analysis

The previous chapter described the use of the multi-method evaluating method, Evaluating User Experience (UE). This chapter discusses the results of an evaluation of the abstraction techniques and the UE study. The first section discusses the results of an evaluation of the abstraction techniques used to reduced the digital evidence dataset. The second section discusses the results of the UE study, specially the usability testing and the results of the Post-study System Usability Questionnaire (PSSUQ). The user performance during the usability testing and the overall satisfaction measurements from the PSSUQ provided insight to the effectiveness of TAIMA's application of Information Visualization (infovis) techniques into the forensics analysis phase of the digital forensics process. While the evaluation of the abstraction techniques provided insight on efficiency.

## 5.1 Abstraction Evaluation

The ability to manipulate large datasets has become essential as data storage volume continues to grow. During a digital forensics investigation, the analyst must identify pertinent files of interest as well as 'evidence' to support the discovery. Ayers argues, high-level abstraction can improve an analyst's effectiveness in identifying 'evidence'[12].

The addition of the abstraction nodes to the data model reduces the number of nodes that need to be processed when requesting data. TAIMA automatically displays system events as discreet items on a graphical timeline with access to traces using tooltip. The benefits of adding the abstraction nodes eliminates the need to conduct manual review of individual low-level nodes to find traces.

One way to evaluate the efficiency of the abstraction queries is to look at

their performance. Table 2 shows the difference in the number of rows required to perform a search with and without the abstraction nodes when conducting a search for the five system events listed in Table 2. For example, a search for Program Installation events searches 199 rows in the database when executed without the use of the abstraction nodes. However, the same search when conducted using the abstraction nodes searches only 28 rows. The efficiency of adding the abstraction node is clearly evident. Across the five events the reduction is an average of 75%. This significant reduction results in less work by the application to fetch data.

**Table 2. Rows Hits per Query**

| Rows | Without Abstraction | With Abstraction | Abstraction/ Without Abstraction (%) |
|---|---|---|---|
| Program Installation | 199 | 28 | 0.14 |
| Power Events | 137 | 72 | 0.53 |
| Program Execution | 4435 | 902 | 0.20 |
| File Download | 42 | 18 | 0.43 |
| Web History | 3824 | 270 | 0.07 |

Another way to understand the efficiency of the abstraction queries is to show the actual execution times with and without abstraction. Table 3 shows actual execution times for each of the five abstraction queries.

Table 3. Query Search Time (milliseconds)

| Time Comparison | Without Abstraction (ms) | With Abstraction (ms) | Abstraction/ Without Abstraction (%) |
|---|---|---|---|
| Program Installation | 193 | 5 | 0.0260 |
| Power Events | 16 | 5 | 0.3100 |
| Program Execution | 4295 | 30 | 0.0007 |
| File Download | 3025 | 7 | 0.0023 |
| Web History | 3116 | 18 | 0.0006 |

From both tables it clear that conducting searches using the abstraction nodes is more efficient than searching without using the abstraction nodes.

## 5.2 Data Analysis

### 5.2.1 Evaluating User Experience (UE) results.

UE usability testing provided measurements to evaluate the processes, visualization and the fundamental design principles of TAIMA. The evaluation measures task performance (time taken to complete the tasks.), accuracy (number of hacking software found out of 6.) and user satisfaction ratings (subjective ratings in a post-task survey that describes how the participants felt about the system.). Combined the measures provides evidence of how TAIMA minimizes the difficulties associated with forensic analysis and and provides relevant information in a easy to understand display.

### 5.2.2 Performance.

Performance measurements, collected as time on task, provides a value by which to evaluate the overall effectiveness of the tool in helping to analyze and display digital evidence. Before starting the session each participant was instructed to complete the task within a 30 minutes time limit. In all five session the investigator stopped the session after an hour. However, during all five sessions the investigator observed the participants being presented with all six hacking software after entering a time window on their first attempt. Additionally, a review of each participant's journal revealed the six hacking software was discovered by all participants.

During a discussion after the session the participants expressed it did not take them long to found all six. One participant talked about wanting to do some additional exploration of the tool's capabilities. Another participant tried to find a way to view contents of files. This indicates participants may have misunderstood how to complete the task or the instructions in the task description should be administered differently in future research.

### 5.2.3 Accuracy.

The accuracy metric of the performance measurement evaluates if the infovis provided the correct information to complete the task. Accuracy was an important metric to collect because not finding any hacking software would indicate serious design flaws. To collect the measurement each participant was instructed to keep a journal of which software on the timeline was used for hacking. At the end of the session the journal was turned in to the research investigator.

In all five session the participants found the six hacking software applications. This result was expected as the timeline clearly displayed discreet system events.

Additionally, given access to the internet to search unknown files the participants felt confident identifying applications that were used in the simulated hacking case. Additionally, the tooltip provided full path execution that helped provide additional corroborating information.

### 5.2.4 Usability.

The usability rating was determined from answers to the PSSUQ presented to the participants after completing the analysis task [60]. The rating provides an overall indication of a participant's satisfaction of the tool and the tool's usability regarding the analysis of digital evidence [60].

The results from the 16-item PSSUQ are shown in Figure 28. The 7-point rating scale ranged from 1 (Strongly disagree) to 7 (Strong agree) (Note: Higher scores indicate better usability.).

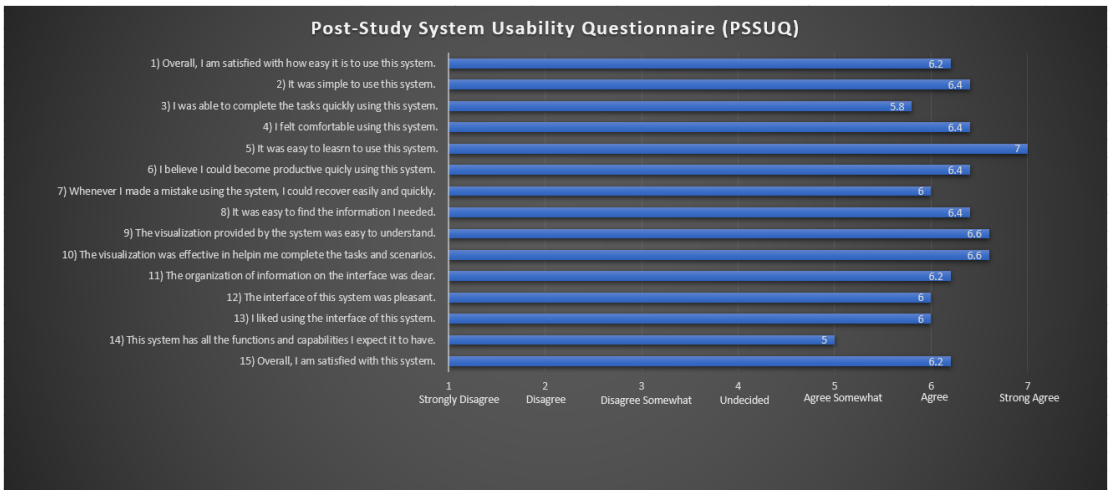



**Figure 28. Post-study System Usability Questionnaire (PSSUQ)**

In a more detailed analysis, item 5 (*It was easy to learn to use this system.*) and item 8 (*It was easy to find the information I needed.*) received the highest scores. This highlights the effectiveness of the overall design and visualization technique of the application display. Moreover it is a testament to the intuitiveness

and visualization techniques to display data in a understandable format. More importantly, it highlights that the visualization techniques implemented can be used to assist an examiner establish a overview of the activity on system during a digital forensics investigation.

The items that received the two lowest scores were 3 (*I was able to complete the tasks and scenarios quickly using this system.*) and 14 (*This system has all the functions and capabilities I expect it to have.*) with scores 5.75 and 5.5 respectively.

During the sessions the research investigator observed all five participants used a time interval of 27 Aug 2004 12:00:00 AM - 27 Aug 2004 12:00:00 PM, which displayed the six hacking software. The low scores could be due to the participants not understanding the goal of the task. Question 14 low score could be as result of the participants wanting to do more analysis than was expected to meet the goal of the experiment. Two of the participants ask if they could see the contains of text files. Viewing contents of files was not apart of the task.

The second part of this analysis examines the three sub-scores of the PSSUQ: System Quality (the average of items 1-6), Information Quality (the average of items 7-12), and Interface Quality (the average of items 13-16). The overall satisfaction score is the average of the three sub-scores [60].

The following is a break-down of the sub-categories:

1. System Usefulness (average of items 1-6)

    (1) Overall, am satisfied with how easy it is to use this.

    (2) It was simple to use this system.

    (3) I was able to complete the tasks and scenarios quickly using this system.

    (4) I felt comfortable using this system.

(5) It was easy to learn to use this system.

(6) I believe I could become productive quickly using this system.

2. Information Quality(average of items 7-12)

(7) Whenever I made a mistake using the system, I could recover easily and quickly.

(8) It was easy to find the information I needed.

(9) The visualization provided by the system was easy to understand.

(10) The visualization was effective in helping me complete the tasks and scenarios.

(11) The organization of information on the interface was clear.

(12) The interface of this system was pleasant.

3. Interface Quality (average of items 13-15)

(13) I liked using the interface of this system.

(14) This system has all the functions and capabilities I expect it to have.

(15) Overall, I am satisfied with this system

In an evaluation of the PSSUQ, Lewis [62] established strong correlation between System Usefulness and task completion; and between accuracy and Information Quality. This correlation is reflected in this research and illustrated in Figure 29. Item 5 (It was easy to learn to use this system.) in System Usefulness received the highest rating followed by items 9 (The visualization provided by system was easy to understand.) and 10 (The visualization was effective in helping me complete the tasks and scenarios.) which are included in the Information Quality average.

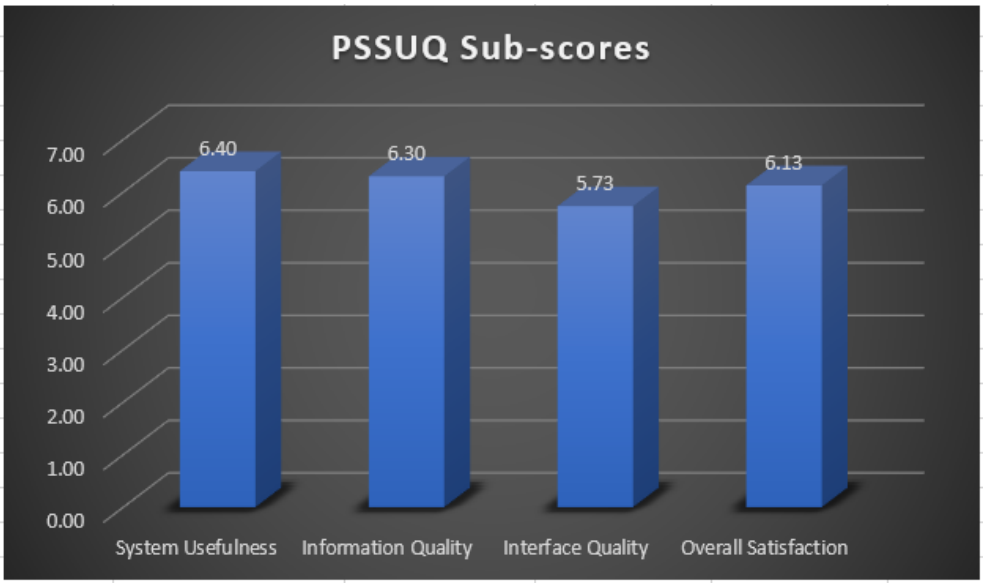



**Figure 29. Post-study System Usability Questionnaire (PSSUQ) subscores. Note: Higher scores denotes better usability.**

Results from the PSSUQ sub-scores revealed that the participants were overall satisfied with the usability of TAIMA indicated by a 6.23 rating out of 7 (89%). The overall satisfaction score is an average of three sub-scores. The highest average among the three sub-scores was 'System Usefulness'. The lowest average was for 'Interface Quality' which includes item 14 (*This system has all the functions and capabilities I expect it to have.*) that received the lowest rating among the individual items. The low score was expected because TAIMA is a prototype. The emphasis during development was on data reduction and displaying accurate information.

System Usefulness received the highest scores of the three sub-scores which suggests the visualization implemented by TAIMA was accurate and help users found the information they were looking for in a timely manner. All participants like that they did not have to do alot of searching to find revelant artifacts (item 8). Feedback also revealed they appreciated all the important information presented to them on one screen (item 10 and 11). Additionally, all the participants found

TAIMA to be easy to use, due to straightforward controls and an intuitive display (item 5).

The Interface Quality sub-score had the lowest score (5.73). In the Interface Quality sub-category item 14 (*This system has all the functions and capabilities I expect it to have.*) received the lowest score among the individual items.This result was expected due to TAIMA being a prototype.

The results of the PSSUQ shows that forensic analysis clearly benefits from inforvis and a display that uses high-level abstraction to present system events.

This section discusses the comments provided by the participant in the feedback section of the questionnaire.

*"I definitely thought the visualization was useful as it made observing that activity quite easy and fast. This would be extremely beneficial."*

The comment highlights the efficiency and effectiveness of using the application to complete the digital forensics analysis task. The infovis techniques and the use of the timeline made navigation easy. Additionally, the integration of abstraction and infovis techniques reduced the amount of information displayed on the screen which made identifying particular events fast. Furthermore, this highlights the advantage gained from eliminating unnecessary information and only displaying important information.

*"This is a very good idea and after a few tweaks could be very usable"*

This feedback highlights the potential of the techniques and implementation used to build TAIMA. This is important as it shows TAIMA is considered a practical approach that adds efficiency to digital forensics analysis phase of a digital

forensics investigation.

*"very easy to understand system."*

This highlights the ease of understanding the visualization provided by TAIMA. The design objective was to implement a simplistic design. Industry standard tools are well known for overcrowding the display. Additionally, this feedback indicates the effectiveness of displaying high-level abstraction displayed in a sequential order.

*"The timeline of events was useful. Matching executables to .lnk instances was useful."*

This feedback illustrates the effectiveness of including a timeline as the display apparatus to provide vital information. Using the temporal property of the data as a way to filter the data only displayed the requested information. The feedback also highlights the usefulness of the tooltip displaying trace information. This technique gives the user the location of the supporting evidence of the system event without having to use another tool or go to another screen. Additionally, not displaying the information until requested is another form of filtering that unclutters the display.

**Summary**

Analysis of the performance and satisfaction measures provides evidence to support how effective TAIMA could be as a tool for the analysis and presentation of digital evidence. The analysis also provided answers for the research questions:

1. *What Information Visualization practices reduce the digital forensics challenge of evidence volume and complexity within the digital forensics analysis*

*process?*

Performance (time taken on task) and accuracy measurements illustrates that the data transformation processes and abstraction techniques assists examiners to evaluate vast amounts of heterogeneous digital evidence accurately and in a timely manner. This suggests that the integration of exploratory infovis techniques as well as abstraction techniques are contributing factors to the effectiveness and efficiency of TAIMA.

Moreover, it underscores that the combination of the infovis techniques combined with abstraction techniques enables reduces the digital forensics challenge of evidence volume and complexity within the digital forensics analysis process.

2. *To what degree does the use of a graphical timeline integrated with Information Visualization best-practices support the digital forensics analysis processes?*

   The overall satisfaction rating from the PSSUQ demonstrated that a graphical timeline, abstraction techniques and best practice infovis support the digital forensics analysis processes and help examiners gain a better understanding of a digital evidence collection. The high usability rating represents how participants graded TAIMA in terms of ease of use, ease of learning, simplicity, effectiveness, information and the user interface. These elements supports the main research goal of minimizing the impact of the key challenges of data volume and complexity in digital forensic analysis.

While TAIMA has been successful so far in providing novel solutions to mitigate the complexities of digital forensics analysis, it is important to remember this is a prototype still in the design phase. Further validation and testing with

bigger datasets need to be accomplished.

# VI. Conclusion

An essential undertaking during a digital forensics investigation is to reconstruct past events during forensics analysis. Digital forensics analysis and event reconstruction have become a difficult task over the past decade with the rapid evolution of digital technologies and their omnipresence in daily life. Consequently, finding the right tool for digital forensics analysis is difficult. The majority of digital forensics industry-standard tools use text-based timelines. Conducting forensics analysis using a text-based timeline for digital forensics analysis is a manual, labor-intensive process.

Previous work proposes Information Visualization (infovis) and a graph-based timeline as a solution to mitigate the challenges of conducting a forensics analysis using a text-based timeline. The graphics enable examiners to easily identify correlations between system events by reducing the amount of data to review while still providing an overview perspective of the dataset. Novel Analysis Integration Management Application (TAIMA) integrates infovis technology with a graphical timeline to generate a graphical user interface (GUI) prototype application. This research illustrated that TAIMA reduces the challenges due to the increasing complexity and volume of modern digital devices.

TAIMA uses a graphical timeline to enhance event reconstruction by providing a graphical timeline using temporal event abstraction and visualization techniques. The graphical timeline displays a vast amount of heterogeneous data in sequential order based on their temporal attributes. Furthermore, TAIMA uses temporal abstraction techniques to transform vast amounts of low-level system events into a significantly smaller number of high-level events.

## 6.1 Results

The overall goal of this research was to improve digital forensics analysis effectiveness and efficiency via a graphical timeline and accelerate digital evidence detection. The goal was met as TAIMA minimized the manual, labor-intensive practices needed during forensic analysis. The multi-method research approach provided answers to research questions that helped satisfy the research objective and goal.

Firstly, the workflow demonstration illustrated how the information visualization (infovis) and temporal event abstraction techniques provided answers to the first research question found in Chapter 1:

> *(1) What Information Visualization (infovis) practices reduce the digital forensics challenges of evidence volume and complexity within the digital forensics analysis process?*

The demonstration showed that TAIMA temporal event abstraction substantially reduced the rows and nodes when searching the database. The compressed datastore enabled the user to view an overview of system events and temporal proximity of system events that occurred within a specific timeframe. Additionally, with a direct link to low-level traces, the examiner can evaluate the circumstance surrounding the system events and build a theory about the use of the system. Without infovis examiners are left to analyze vast amounts of data among which only snippets are of importance.

Secondly, the pilot study usability test provided answers to the second research question:

> *(2) To what degree does the use of a graphical timeline integrated with information Visualization best-practices support the digital analysis process?*

TAIMA received an 89% overall satisfaction rating. All five participants completed the analysis task successfully by accurately identifying potential hacking software. The results from the post-task questionnaire shows that the study participants consider TAIMA to be a practical application. This indicates that the use of a timeline in digital forensics tools allows investigators to establish situational awareness of system events on a hard drive in a relatively short period of time with high accuracy.

## 6.2 Limitations

There were three main limitations observed throughout the research. Firstly, the scope of the study was limited to finding only executables and web history. This helped reduce the time of the usability testing sessions; however, it also limits the testing of the application. Future research should present scenarios that include more suspicious files of interest and more complex scenarios.

Secondly, due to limited resources and proximity, the participants conducted the evaluation remotely. Ordinarily, usability laboratories are used to conduct usability tests. The lab environment provides an area that allows the investigators to observe the participants. To reduced the effect of this particular limitation the participants were asked to turn in an investigative journal that included their plan of action to complete the assigned task.

Finally, the fictitious case used in the controlled experiment simulated case data and not actual crime scene systems. The use of the fictitious data was an effort to controlled the size of the dataset and exposure of personally identifiable information. Evaluating the performance of TAIMA with a larger dataset is an important next step. The evaluation will provide important measurements to identify critical design or implementation improvements. As the size of the image

files increased the infovis and abstraction techniques is expected to sustain the high accuracy and performance.

### 6.3 Future Work

The usability study carried out to evaluate TAIMA was the logical first step in evaluating the viability of the graph-based timeline for digital forensics analysis. As a pilot study, only a small number of digital forensics experts were targeted to participate. Future research needs to increase the participant pool. The increased participants' pool would improve the confidence in the results and reduce the effects of extremely high or low outcomes.

TAIMA does not provide export or printing capabilities. Currently, the user would have to use print-screen to generate any form of reports. This is not a desirable solution as an essential part of the digital forensics process is to report findings.

Giving the user more customizing option regarding which artifacts to filter from view can make the application more effective by further reducing the already filtered data. This customizing is another filter to reduce the number of events on the timeline making TAIMA more efficient and effective by giving the user more control over what they want to review.

Finally, the control experiment was limited in scope. The fictitious hacking case data used for usability testing was simulated. The test imaged was pre-loaded with predetermined system events of only one user. Future research should conduct more robust testing with a larger image and more complex systems events that more closely simulate a real-world hard drive.

This exploratory study demonstrated a strong performance by TAIMA, mainly in how accurately it was able to assist expert digital forensics specialists in iden-

tifying key system events from a relatively large dataset. Further exploration is highly encouraged.

# Appendix A. Approval

**DEPARTMENT OF THE AIR FORCE**
**AIR FORCE RESEARCH LABORATORY**
**WRIGHT-PATTERSON AIR FORCE BASE OHIO 45433**

MEMORANDUM FOR AFIT/ENG (GILBERT PERTERSON)

FROM: 711 HPW/IR

SUBJECT: IRB Approval for the Use of Human Volunteers in Research

1. Protocol Title: Digital Forensics Graphical User Interface Visualization Platform

2.

| Protocol Number | Protocol Version | Risk Level |
|---|---|---|
| FWR20190017H | V1.00 | Minimal Risk |

3.

| Approval Date | Expiration Date | Re-approval Request Due by: |
|---|---|---|
| 15 January 2019 | 14 January 2020 | 14 December 2019 |

4. This study received Expedited Review using regulatory category: 32CFR219.110 (b)(7)

5. Summary: This research will study the effectiveness and efficiency of the graphical user interface, enhanced with a graph-based timeline visualization for temporal event reconstruction. This study will recruit 10 AFOSI computer crimes investigators with experience in conducting digital forensic investigations and analysis.

6. A waiver of documentation of consent has been granted for this research project as it meets the criteria outlined in 32 CFR 219.117 (c).

7. All inquiries and correspondence concerning this protocol should include the protocol number and name of the primary investigator. Please contact the 711 HPW/IR office using the organizational mailbox at AFRL.IR.ProtocolManagement@us.af.mil or by calling 937-904-8094 [DSN 674].

LONDON.KIM.ELIZABETH.1155556370 Digitally signed by LONDON.KIM.ELIZABETH.1155556370 Date: 2019.01.16 15:01:01 -05'00'

KIM E. LONDON, JD, MPH
Chair, AFRL IRB

# Appendix B. Institutional Review Board Memorandum

**Digital Forensics Graphical User Interface Visualization Platform**

1. **Principal Investigator**
   Dr. Gilbert Peterson/Civ/Professor, AFIT/ENG, (937) 255-3636, gilbert.peterson@wpafb.af.mil

2. **Associate Investigators**
   1stLt Nikolai Adderley/Student, AFIT/ENG, 253-777-8204, nikolai.adderley@afit.edu

3. **Research Monitor**
   N/A – Minimal Risk Study

4. **Facility/Contractor**
   **4.1. Sponsor: Department of Defense Cyber Crime Center**
   **4.2. Funding Source and Funding Amount: none**
   **4.3. Contract #/CRADA #/Cooperative Agreement #: none**
   **4.4. Activity location(s) (where activity will be conducted): online**

5. **Conflicts of Interest**
   None.

6. **Background Information and Scientific Rationale**
   A significant part of every digital forensic investigation is knowing when events occurred and the time those events occurred, as known as event reconstruction [5]. Over the last decade, advances have been made in the early stages of the digital forensics lifecycle (acquisition, preservation and searching), but, unfortunately, developments to assist examiners during the analysis stage of the cycle, where event reconstruction is performed, continue to struggle to keep up[6]. The rapidly evolving digital technology and the significant increase in the volume of data generated and stored by these devices have made analysis and event reconstruction difficult [7]. Industry standard digital forensics applications, such as EnCase or FTK, has streamlined finding data on the target machine [8]. However, the tools have been reported to lack graphical displays of temporal information of files needed to conduct an efficient event reconstruction analysis [9]. As a result, the investigator must look through all the collected data in detail using manual, ad-hoc processes [3].

   One answer to those issues is a visualization that reduces information overload by providing an overview of the data, enables focused analysis through data filters, and increases comprehension of the dataset [2]. Visualization uses graphics to highlight patterns, sequences and relationships within a dataset [10]. The main focus of visualization in digital forensics is information visualization used to enhance the exploration of datasets [2]. One area of research showing promising results is the field of event reconstruction using graph-based timeline. Event reconstruction is defined [9] “as a process of taking as input a set of events and outputting a timeline of the events describing the case.”

   A literature review revealed there are a varying number of tools with event reconstruction capabilities [8]. Olsson and Boldt [3] created the prototype called CyberForensic TimeLab (CFTL). CFTL combines date and time extraction techniques while storing indexed entries into a centralized database. A GUI allows users to filter and sort data based on date and time.

Users can focus their analysis down to only a specific time interval of interest based on intelligence from other investigative techniques. However, CFTL is not open source.

Osborn et al. [2] developed a software based on the Explore, Investigate, Correlate (EIC) framework. The authors claimed the software provides three advanced capabilities via a GUI. Firstly, the software renders a high-level view of the dataset that can be examined using filters and tracing. Additionally, the tool displays the links between events in the dataset using a 'correlative visualization technique'.

In short, the literature reviewed revealed visualizations in digital forensics will always face the challenge of presenting a vast number of events in a user-friendly GUI with explorative capabilities and features. This research is, therefore, an effort in the development of a digital forensics investigation tool that employs both visualization and data exploitation techniques for easier event reconstruction

## 7. Study Objective(s) and Purpose

### 7.1. Purpose:

The purpose of this research is to conduct a qualitative usability evaluation of a digital forensics graphical user interface (GUI). This research will study the effectiveness and efficiency of the GUI, enhanced with a graph-based timeline visualization for temporal event reconstruction.

A multifaceted procedure combining experimental research strategies in the area of information visualization evaluation was chosen following a literature review on evaluating visual data analysis and reasoning (VDAR) [1]. As a result of analysis of user-centered design methods and consideration of the research goals, two approaches were selected based on their combined strengths: survey questionnaire (SQ) and usability testing (UX).



### 7.2. Primary Objective:

The motivation for this research is rooted in need to simplify the analysis of digital forensics data. Some of the questions this research will try to answer are documented problems associated with the complexities associated with forensic data analysis and temporal event reconstruction [2, 3, 4]:

- Does the GUI support data exploration?
- What knowledge is gained about the dataset from using the visualization?
- Does the platform support the research objective and interactive examination of the data?
- How does the platform support the analysis phase of digital forensic examination?

### 7.3. Secondary Objective(s):

The findings of this research will help in the design and development of digital forensics visualization platform to reduce the tedious and manual analysis processes examiners experience during a digital forensics investigation. This framework is a viable alternative to the commercially available digital forensics tools that only have limited or non-existent timeline analysis. The framework design and configuration can be used by USAF agencies and other organizations that conduct digital forensics.

## 8. Study Design

### 8.1. Description of Study Design:

This qualitative study aims to address problems associated with streamlining procedures essential to making use of visualization for effective event reconstruction analysis. The motivation behind this research is to develop a user-centered, prototype framework focused on using graphics to automate the visualization of temporal information to make event reconstruction more efficient.

The user will perform a set of tasks using the prototype digital forensics graphical user interface (GUI) to identify particular files of interest related to a notional criminal case. The GUI will provide an interactive, graph-based, visualization using only the temporal property of files from a hard drive image in which normal, as well as malicious behavior, will be performed. The sessions and its corresponding tasks are described in detail in the appendix.

The participants will be provided with access to the interface, supporting documentation and training manual. They will then be provided the tasks either in writing or as a digital copy and given a time limit to complete all the tasks. Data from the user interaction with the GUI will be recorded, including total task completion, and error-rate. Participants will be identified by a unique identifier that will not be associated with their name.

The digital forensics tasks are designed to simulate the analysis phase of a digital forensic investigation. The assigned task will require the participant to use the rendered visualization and inspect the timestamp data to identify events of interest. See the appendix for details on the specific tasks.

The study will take participants roughly one hour to one and half hour to complete. The sessions will be scheduled according to the individual participant's availability. The entire study will take one month for all participants to complete.

After completing the task, the participants' feedback will be captured via questionnaires (refer to appendix for specific questions). To obtain participants' reactions to the GUI a post-test questionnaire will be administered after completing the task described in the scenario. The questionnaire includes demographic questions as well as questions regarding feedback from the user about their experience with the GUI. It combines both open-ended, free-text answers along with some rating answers (i.e. such as asking for the perceived task difficulty on a scale of 1-7) to enable quantitative and qualitative data analyses.

Data analysis will include responses from the questionnaire along with the calculation of: the success rate (number of task/total number of task times 100) and the average (mean) time taken to complete the task. The data will then be analyzed by the research investigator to identify fundamental design strategy strengths and weaknesses. The post analysis will focus on the problems the subjects faced in completing the tasks and identification of solutions to implement in future research of this nature.

## 9. Subject Selection

### 9.1. Inclusion Criteria:

A subject who has met all of the following criteria is eligible for participation in the study:

Computer Crimes Investigator (CCI) career field member within the USAF Air Force Office of Special Investigations (AFOSI) and individuals with similar background will be asked to participate.

The participants must have a background and experience in digital forensics.

Age range will be 21 and up.

**9.2. Exclusion Criteria:**

A subject who meets any of the following criteria is disqualified from participation in the study:

No special subjects will be involved (45 CFR 46 subparts B-D).

Subjects that are not USAF AFOSI CCI Special Agent or similar with a background in digital forensics.

**9.3. Recruitment Plan**

This study will enroll up to 10 participants. Participants will be Special Agents from the United States Air Force (USAF) Office of Special Investigations (AFOSI) 3rd Field Investigations Squadron, Lackland AFB, Texas, and 2nd Field Investigations Squadron, Joint Base Andrews, Maryland. The participants will be AFOSI agents, both military and civilian, who are computer crimes investigators with experience in conducting digital forensic investigations and analysis. They also assist other agents analyze data and evaluate its significance to the investigation. Recruitment of personnel will be via email or word of mouth as non-paid volunteers. Furthermore, recruitment of personnel will not include a superior or a person in their chain of command in order to prevent coercion.

Subject recruitment emails or webpages will include the following content:

"The Air Force Institute of Technology (AFIT) is conducting a study in which participants will perform computer-based tasks using digital forensic tool graphical interface enhanced with information visualization technologies. The experiment is structured to simulate the analysis phase of a digital forensic investigation. A variety of data regarding task performance will be acquired during the trials, and demographic data will be collected as well.

The main goal of this study is to investigate how a digital forensics tool interface enhanced with visualization techniques may improve and examiner's capabilities in identifying digital evidence. Participation in this study is voluntary and there is no compensation. However, participation in the study will allow you to take part in important research about digital forensic and help the investigators detect the effect of a graphical timeline on functions and performance. Volunteers will be asked to participate in the computer-based experiment. Participants will work on task for up to 30 minutes. This research project has been approved for the use of human subjects by the Air Force Research Laboratory's Institutional Review Board in accordance with AFI 40-402 and AFRLI 40-402."

**9.4. Consent Plan**

Potential participants will be provided an informed consent document by the primary or associate investigators and be allowed to review and get responses on any questions they have. They will receive a copy of the consent form prior to undertaking the experiment.

WAIVER OF DOCUMENATION OF CONSENT APPLICATION:

The investigator requests a waiver of signed consent (i.e., consent signature). This is requested because the only record linking the subject and the research would be the consent document and the principal risk would be potential harm resulting from a breach of confidentiality. Each subject will be asked whether the subject wants documentation linking the subject with the research, and the subject's wishes will govern.

**9.5. Compensation**

There are no plans to provide compensation.

## 10. Experimental Plan

### 10.1. Equipment:

The only equipment the subjects will interact with is the computer. The subjects will primarily use the computer for accessing the GUI.

## 11. Risk/Benefit Analysis

### 11.1. Benefits:

There is no direct benefit to the subjects.

### 11.2. Risks:

This study presents limited physical risk to the subjects' typical office work (e.g. eye strain, repetitive strain injury).

The study presents no known psychological risks that the research team is aware of.

## 12. Statistical Consideration and Plan

### 12.1. Sample Size (Power analysis):

As a pilot study, the anticipated sample pool consists of four to eight individuals. No statistical analysis can be conducted on such a small pool.

## 13. Safety Monitoring and Reporting

This study presents limited physical risk to the subjects' typical office work (e.g. eye strain, repetitive strain injury). Therefore, we believe safety monitoring such as an on-site medical observer is not necessary.

## 14. Confidentiality

Participant's name, rank, sex and other personal information will NOT be asked or recorded. However, subjects will be asked about their years of digital forensics experience. Each subject will be assigned a numeric ID which will be used to label all data they produce.

Participants will be assigned a random identification (ID) number in the study, which will be used as the primary identifier. All data will be reported and stored only by participant ID number. Furthermore, the data will be reported only as aggregates and only for the research purpose. A master list linking participant ID numbers with the names of the participants and the Informed Consent documents will be filed separately from the rest of the experiment materials, in a separate secure folder on AFIT network, accessible only by the PI or specified associate investigators. When no longer needed for research purposes, identifying information and data will be destroyed in a secure manner. Participants will never be identified by name in any report or publication.

**15. Data Management/ Data Sharing Plan**

There are no plans to send the data to a research data repository.

**16. References**

[1] H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale, "Seven Guiding Scenarios for Information Visualization Evaluation Seven Guiding Scenarios for Information Visualization Evaluation," *Tech. Rep. DCS Univ. Calgary*, p. 17, 2011.

[2] G. Osborne, B. Turnbull, and J. Slay, "Development of InfoVis Software for Digital Forensics," 2012.

[3] J. Olsson and M. Boldt, "Computer forensic timeline visualization tool," *Digit. Investig.*, vol. 6, no. SUPPL., 2009.

[4] Y. Chabot, A. Bertaux, C. Nicolle, and T. Kechadi, "Automatic timeline construction and analysis for computer forensics purposes," *Proc. - 2014 IEEE Jt. Intell. Secur. Informatics Conf. JISIC 2014*, pp. 276–279, 2014.

[5] B. D. Carrier and E. H. Spafford, "Defining event reconstruction of digital crime scenes.," *J. Forensic Sci.*, vol. 49, pp. 1291–1298, 2004.

[6] D. Edwards, "Computer Forensic Timeline Analysis with Tapestry Computer Forensic Timeline Analysis with Tapestry Computer Forensic Timeline Analysis with Tapestry 2."

[7] G. Mohay, "Technical challenges and directions for digital forensics," *Proc. - First Int. Work. Syst. Approaches to Digit. Forensic Eng.*, vol. 2005, pp. 155–161, 2005.

[8] G. Palmer, "the first Digital Forensic Research Workshop," *First Digit. Forensic Res. Work.*, no. 1, pp. 15–18, 2001.

[9] Y. Chabot, A. E. Bertaux, C. Nicolle, and M.-T. Kechadi, "A complete formalized knowledge representation model for advanced digital forensics timeline analysis," 2014.

[10] W. Liao and D. Ph, "Data Visualization in the Geosciences," *Technometrics*, vol. 47, no. 3, pp. 382–382, 2005.

**17. Attachments**

- Informed Consent Document.
- Current Curriculum Vitae of investigators.
- Questionnaire.

**Abbreviated Informed Consent**
**Digital Forensics Graphical User Interface Visualization Platform**
**FWR20190017H**

You are being asked to participate in a research study. The purpose of this research is to evaluate a prototype digital forensics graphical user interface (GUI), enhanced with a graph-based timeline visualization. The visualization provides novel techniques to reduce information overload faced by examiners when attempting to perform event reconstruction with digital evidence.

Outcomes for this human-computer interaction experiment are twofold:

A. Evaluate the effectiveness of the prototype GUI, integrated with a graphical timeline visualization, for data exploration and analysis.

B. Reduce the reliance on manual techniques, considered as primary tasks, from the digital forensics investigation process to increase discovery of evidentiary artifacts.

The expected length of your participation is approximately one hour.

If you participate in this research, as a subject matter expert, you will be asked to perform a digital forensics evidence analysis for a notional criminal case. The task is intended to simulate the analysis phase of a digital forensics investigation. During the session, you will be asked to identify and record potential digital artifacts of interest in a simulated digital forensics investigation. A graph-based timeline visualization displayed in a GUI is to be used. The interface records your interactions as you complete the task. Upon completion of the task, a questionnaire will ask you to identify the problems you encountered, what was easy, hard, and any recommendation for improvements. You may take as many breaks as you need during the session.

Reasonably foreseeable risks to your participation are:
Risks for most participants will be similar to risks experienced by a typical desktop computer user. However, because you will need to be attentive to the display over an extended period of time you may experience a slightly greater risk of eye fatigue and dry eyes.

Discomforts may consist of eye, wrist and hand strain typical of office/computer work. To minimize and/or alleviate symptoms, frequent breaks away from the desk are suggested.

You are not expected to benefit directly from participation in this research study. This study's results will benefit future research by providing human-computer interaction metrics that will be used in future design and development of graphical timeline visualization technology used in digital forensics to increase and improve examiners' efficiency and accuracy.

Your decision to participate in this research is voluntary. You can discontinue participation at any time without penalty or loss.

The researchers will take the following precautions to maintain the confidentiality of your data:
The researchers will not collect any identifiers linked to you. Your responses will be distinguish by an integer, and no participant identifiers will be included in any publications. Electronic data will be password-protected.

The data may be accessed by the Department of Defense for auditing purposes.

If you have questions regarding the study, contact the Principal Investigator: Gilbert L. Peterson gilbert.peterson@afit.edu or 1stLt Nikolai Adderley nikolai.adderley@afit.edu. If you have questions regarding your rights as a research subject, contact the AFRL IRB: 937-904-8100 or afrl.ir.protocolmanagment@us.af.mil.

## Questionnaire

**Participant Name:**

**Date:**

**Purpose**

The purpose of this research is to conduct a quantitative usability evaluation of a prototype digital forensics graphical user interface (GUI). This research will study the effectiveness and efficiency of the GUI, enhanced with a graph-based timeline visualization, in regards to temporal event reconstruction. The visualization objective is to provides novel techniques to reduce information overload faced by examiners when attempting to perform event reconstruction with digital evidence. When developed and implemented correctly, visualization enables unrestricted access and interaction to large datasets. Providing examiners with direct interaction is critical in facilitating the rapid discovery of useful information within large data sets.

**Background Questions**

1. What is your primary duty?
2. How many years of experience do you have performing digital forensics examinations?
3. Please list any professional digital forensics certifications you have?

**End-of-Session Survey**

*INSTRUCTIONS:* Please rate the degree to which you agree or disagree with each of the following statements related to the assignment you were asked to complete and your experience with GUI.



| | | **Strongly Agree** | | | | | | **Strongly disagree** |
|---|---|---|---|---|---|---|---|---|
| | **Questions** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| 1 | Overall, am satisfied with how easy it is to use this system | | | | | | | |
| 2 | Overall, I am satisfied with this system | | | | | | | |
| 3 | I was able to complete the tasks quickly using this system | | | | | | | |

| | | Strongly Agree | | | | | | Strongly disagree |
|---|---|---|---|---|---|---|---|---|
| | **Questions** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| 4 | Overall, I am satisfied with this system | | | | | | | |
| 5 | I believe I could become productive quickly using this system | | | | | | | |
| 6 | Whenever I made a mistake using the system, I could recover easily and quickly | | | | | | | |
| 7 | The tutorial provided with this system was easy to understand | | | | | | | |
| 8 | It was easy to find the information I needed | | | | | | | |
| 9 | The visualization provided by the system was easy to understand | | | | | | | |
| 10 | The visualization was effective in helping me complete the tasks and scenarios | | | | | | | |
| 11 | The organization of information on the interface was clear | | | | | | | |
| 12 | The interface of this system was pleasant | | | | | | | |
| 13 | I liked using the interface of this system | | | | | | | |
| 14 | This system has all the functions and capabilities I expect it to have | | | | | | | |

**Survey Questions**

1. What aspects of the visualization did you find most useful?

2. What aspects of the visualization needs improvement?

3. What other interactions with the data would you like to have?

4. What did you have a hard time understanding or using?

5. If you would like, please leave any further comments below.

# Appendix C. Study Instructions

*STUDY INSTRUCTIONS*

**Background:**

Today, most crimes being committed utilize some form of digital device. As a result, law enforcement agencies are collecting an increasing amount of digital evidence in conjunction with an investigation. Using industry-standard digital forensics tools, examiners are forced to use manual, labor-intensive efforts to extract and identify digital evidence from the collected digital evidence. This research studies how an enhanced digital forensics graphical user interface (GUI), integrated with visualization techniques can improve an examiner's analysis efforts to efficiently detect potential digital evidence.

**Purpose:**

The purpose of this research is to conduct a preliminary usability study on a prototype digital forensics GUI integrated with graph-based timeline visualization. The goal is to identify to what extent the prototype supports an examiner's analysis efforts to detect potential digital evidence.

The study involves SME participants to:

- Evaluate the effectiveness of the prototype graph-based timeline visualization for examination of digital evidence and the insight it provides to the users.
- Examine the effectiveness of the GUI in mitigating manual processes associated with exploring vast amount of digital evidence.
- Obtain the participants' feedback regarding the prototype.

Studies have shown that visualization integrated into digital forensic tools increases the examiner's ability to identify suspicious evidence rapidly[1,2,3]. The GUI displays an abstracted view of the extracted digital evidence computer files to the examiner based on the timestamp property of a hard drive artifacts.

**Your Role:**

In this hypothetical case, on 27 Aug 2004, a notebook computer, a wireless PCMCIA card and an external homemade 802.11b antennae were found abandoned. The incident response team believe this equipment was used for hacking purposes. The team has knowledge that he hacking suspect, G=r=e=g=S=c=h=a=r=d=t (The equal signs are just to prevent web crawlers from indexing the name; there are no equal signs in the name.) goes by the online nickname of "Mr. Evil". Additionally, some of his associates have said that he would park his vehicle within range of Wireless Access Points (like Starbucks and other T-Mobile Hotspots) where he would then intercept internet traffic, attempting to get credit card numbers, usernames & passwords. The incident response team imaged the computer, extracted the system files and stored them in a database.

As the examiner assigned to the case you were briefed on the details of case by the incident response team. The team is asking you to identify any hacking software use and evidence of their use (path of execution folder). Your report should include significant dates and time of activities related to the hack.

You will take on the role of a digital forensic examiner and perform the digital analysis tasks often performed by a forensic examiner during a digital forensic examination. Your task is to:

- Examine the hard disk drive
- Identify who might be responsible for the hack
- Identify any hacking application used and evidence to support their use
- Establish a timeline for when the hacking activities occurred
- Identify any other files or activity related to the hack that seems suspicious.

Please provide a written report of your findings to the team. Include in the report the timeline, location of system artifacts associated with the hack along with a brief justification for system artifacts you included in the report.

References:


[1] G. Osborne and J. Slay, "Digital forensics infovis: An implementation of a process for visualisation of digital evidence," *Proc. 2011 6th Int. Conf. Availability, Reliab. Secur. ARES 2011*, pp. 196–201, 2011.

[2] R. A. Altiero, "Digital Forensics Tool Interface Visualization," no. 24, 2015.

[3] G. Schrenk and R. Poisel, "A Discussion of Visualization Techniques for the Analysis of Digital Evidence," *2011 Sixth Int. Conf. Availability, Reliab. Secur.*, pp. 758–763, 2011.

# Appendix D. Raw Data

| | Question 1 | Question 2 | Question 3 | Question 4 | Question 5 | Question 6 |
|---|---|---|---|---|---|---|
| **Participants** | | | | | | |
| 1 | 6 | 6 | 6 | 6 | 7 | 6 |
| 2 | 7 | 7 | 6 | 7 | 7 | 6 |
| 3 | 6 | 6 | 5 | 6 | 7 | 6 |
| 4 | 6 | 7 | 6 | 7 | 7 | 7 |
| 5 | 6 | 6 | 6 | 6 | 7 | 7 |
| **Average** | **6.2** | **6.4** | **5.8** | **6.4** | **7.0** | **6.4** |

| | Question 7 | Question 8 | Question 9 | Question 10 | Question 11 | |
|---|---|---|---|---|---|---|
| **Participants** | | | | | | |
| 1 | 7 | 7 | 6 | 6 | 5 | |
| 2 | 6 | 7 | 6 | 6 | 6 | |
| 3 | 6 | 7 | 7 | 7 | 6 | |
| 4 | 7 | 6 | 7 | 7 | 7 | |
| 5 | 4 | 5 | 7 | 7 | 7 | |
| **Average** | **6.0** | **6.4** | **6.6** | **6.6** | **6.2** | |

| | Question 12 | Question 13 | Question 14 | Question 15 | | |
|---|---|---|---|---|---|---|
| **Participants** | | | | | | |
| 1 | 5 | 5 | 4 | 6 | | |
| 2 | 6 | 6 | 6 | 6 | | |
| 3 | 6 | 6 | 5 | 6 | | |
| 4 | 7 | 7 | 7 | 7 | | |
| 5 | 6 | 6 | 3 | 6 | | |
| **Average** | **6.0** | **6.0** | **5.0** | **6.2** | | |

# Bibliography


1. K. Kent, S. Chevalier, T. Grance, and H. Dang, "Guide to integrating forensic techniques into incident response," *National Institute of Standards and Technology: Special Publication 800-86*, pp. 3–1, 2006.

2. G. Osborne, B. Turnbull, and J. Slay, "The 'Explore, Investigate and Correlate' (EIC) conceptual framework for digital forensics information visualisation," *5th International Conference on Availability, Reliability, and Security*, pp. 629–634, 2010.

3. D. P. Groth and K. Streefkerk, "Provenance and Annotation for Visual Exploration Systems," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 6, pp. 1500–1510, 2006.

4. C. Harrell, "What's a timeline," *Journey Into Incident Response*, 2011, [Online]. Available: http://journeyintoir.blogspot.com/2011/09/whats-timeline.html. [Accessed: 23-Jan-2019].

5. M. B. de Carvalho, B. S. Meiguins, and J. M. de Morais, "Temporal Data Visualization Technique Based on Treemap," *International Conference Information Visualisation*, pp. 399–403, 2016.

6. J. Olsson and M. Boldt, "Computer forensic timeline visualization tool," *Science Direct*, vol. 6, 2009.

7. "Neo4j graph platform - the leader in graph databases." [Online]. Available: https:/neoj.com/. [Accessed: Jan. 05, 2019].

8. J. Nielsen, "Why you only need to test with 5 users," *Nielsen Norman Group*, 2000, [Online]. Available: https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/ [Accessed: Nov. 12, 2018].

9. Y. Chabot, A. Bertaux, C. Nicolle, and T. Kechadi, "Automatic Timeline Construction and Analysis for Computer Forensics Purposes," *IEEE Joint Intelligence and Security Informatics Conference*, pp. 276–279, 2014.

10. P. Hitlin, "Internet, social media use and device owernership in u.s. have plateaued after years of growth," *Pew Research Center*, 2018, [Online] http://www.pewresearch.org/fact-tank/2018/09/28/internet-social-media-use-and-device-ownership-in-u-s-have-plateaued-after-years-of-growth/ [Accessed: Nov. 15, 2018].

11. S. L. Garfinkel, "Digital forensics research: The next 10 years," *Digital Investigation*, vol. 7, pp. S64–S73, 2010.

12. D. Ayers, "A second generation computer forensic analysis system," *Science Direct*, vol. 6, pp. S34–S32, 2009.

13. G. Osborne, J. Slay, and B. Turnbull, "Development of InfoVis Software for Digital Forensics," *IEEE 36th International Conference on Computer Software and Applications Workshops*, pp. 213–217, 2012.

14. D. Ogden, "Forensic Science and Forensic Evidence I: Mobile Device Forensics: Beyond Call Logs and Text Messages," *United States Department of Justice Executive Office for United States Attorneys*, vol. 65, no. 1, p. 11, 2017.

15. D. Pati and A. Avinash, "Effective Data Visualization using Tableau," *International Journal of Engineering and Management Research*, vol. 6, no. 5, pp. 306–313, 2016.

16. B. Shneiderman, C. Dunne, P. Sharma, and P. Wang, "Innovation trajectories for information visualizations: Comparing treemaps, cone trees, and hyperbolic trees," *University of Mary, Human-Computer Interacton Lab Technical Report*, vol. 11, no. 2, pp. 87–105, 2012.

17. H. Hibshi, T. Vidas, and L. Cranor, "Usability of forensics tools: A user study," *6th International Conference on IT Security Incident Management and IT Forensics*, pp. 81–91, 2011.

18. X. Du, N.-A. Le-Khac, and M. Scanlon, "Evaluation of Digital Forensic Process Models with Respect to Digital Forensics as a Service," *School of Computer Science University of College Dublin*, 2017.

19. L. Caviglione, S. Wendzel, and W. Mazurczyk, "The Future of Digital Forensics: Challenges and the Road Ahead," *IEEE Computer and Reliability Societies and IEEE Security and Privacy*, vol. 15, no. 6, pp. 12–17, 2017.

20. Pollitt, Mark, Casey, Eoghan, Jaquet-Chiffelle, David-Olivier, Gladyshev, and Pavel, "OSAC Technical Series 0002 A Framework for Harmonizing Forensic Science Practices and Digital/Multimedia Evidence," *Organization of Scientific Area Committees*, 2018, [Accessed: Jan. 07. 2019]. [Online]. Available: http://dx.doi.org/10.29325/OSAC.TS.0002

21. S. Teerlink and R. F. Erbacher, "Improving the computer forensic analysis process through visualization," *Communications of the ACM*, vol. 49, no. 2, p. 71, 2006.

22. G. Hales, "Visualisation of Device Datasets to Assist Digital Forensic Investigation," *Division of Computing Maths, School of Arts, Media and Computer Games*, pp. 1–4, 2017.

23. B. Inglot, L. Liu, and N. Antonopoulos, "A framework for enhanced timeline analysis in digital forensics," *IEEE International Conference on Green Computing and Communications, Conference on Internet of Things, and Conference on Cyber, Physical and Social Computing*, pp. 253–256, 2012.

24. G. Schrenk and R. Poisel, "A Discussion of Visualization Techniques for the Analysis of Digital Evidence," *6th International Conference on Availability, Reliability and Security*, pp. 758–763, 2011.

25. J. Goodall, "Visualization is better! A comparative evaluation," *6th International Workshop on Visualization for Cyber Security*, pp. 57–68, 2009.

26. C. Hargreaves and J. Patterson, "An automated timeline reconstruction approach for digital forensic investigations," *Elsvier Limited*, vol. 9, pp. S69–S79, 2012.

27. Y. Feng, K. Dreef, J. A. Jones, and A. van Deursen, "Hierarchical abstraction of execution traces for program comprehension," *Proceedings of the 26th Conference on Program Comprehension - ICPC '18*, pp. 86–96, 2018. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3196321.3196343

28. GRANDstack, "Getting started with grandstack: Build full stack graph applications with ease," *GRANDstack*, [Online] https://grandstack.io/ [Accessed: Jan. 05, 2019].

29. C. Chuan, "Javascript tutorial: The basics," *Nanyan Technological University, Singapore*, 2015, [Online] http://www.ntu.edu.sg/home/ehchua/ programming/webprogramming/JavaScript_Introduction.html [Accessed: Jan. 05, 2019].

30. G. Osborne and J. Slay, "Digital Forensics InfoVis: An implementation of a process for visualisation of digital evidence," *6th International Conference on Availability, Reliability and Security*, pp. 196–201, 2011.

31. G. Osborne and B. Turnbull, "Enhancing computer forensics investigation through visualisation and data exploitation," *International Conference on Availability, Reliability and Security*, pp. 1012–1017, 2009.

32. R. Carbone and C. Bean, "Generating Computer Forensic Super Timelines Under Linux," *Defense Research and Devlopment Canada-Valcartier Technical memorandum*, pp. 1–136, 2011.

33. S. K. Card and J. D. Mackinlay, "The Structure of the Information Visualization Design Space," *Visualization Conference, Information Symposium and Parallel Rendering Symposium*, pp. 92–99, 1997.

34. B. Shneiderman, "The eyes have it: a task by data type taxonomy for information visualizations," *IEEE Symposium on Visual Languages*, pp. 336–343, 1996.

35. L. Carter, M. Dehart, and J. Gaskin, "The state of infographics," *Mission-Critical Marketing*, pp. 1–8, 2017.

36. B. Shneiderman and C. Plaisant, "Strategies for evaluating information visualization tools," *Proceedings of the BELIV' Workshop Advanced Visual Interfaces Conference, Venice*, pp. 1–7, 2006.

37. E. H. Chi, "A Taxonomy of Visualization Techniques using the Data State Reference Model," *IEEE Symposium on Information Visualization*, pp. 69–75, 2000.

38. B. Nicolau, "Visualization for real time big data," *Master Thesis, Department of Information Systems and Electronic Services, Technical Universit TU Darmstadt, Darmstadt, Germany, 2017.*

39. B. Turnbull and S. Randhawa, "Automated event and social network extraction from digital evidence sources with ontological mapping," *Digital Investigation*, vol. 13, pp. 94–106, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.diin.2015.04.004

40. B. D. Carrier and E. H. Spafford, "Defining event reconstruction of digital crime scenes." *Journal of forensic sciences*, vol. 49, pp. 1291–1298, 2004.

41. K. Gudjonsson, "InfoSec Reading Room Mastering the Super Timeline With log2timeline," *SANS InfoSec Reading Room*, 2010. [Online]. Available: https://www.sans.org/reading-room/whitepapers/logging/mastering-super-timeline-log2timeline-33438

42. E. Casey, "Learn more about temporal analysis," *Science Direct*, 2010, [Online]. Available: https://https://www.sciencedirect.com/topics/medicine-and-dentistry/temporal-analysis/ [Accessed: Nov 12, 2018].

43. A. Prasad and J. Pandey, "Digital forensics," *Uttrakhand Open University*, pp. 1–227, 2016.

44. R. A. Altiero, "Digital Forensics Tool Interface Visualization," Doctoral dissertation, Nova Southeastern University, NSU Works, Graduate School of Computer and Information Sciences, Fort Lauderdale, FL, 2015.

45. J. Slay and F. Schulz, "Development of an Ontology Based Forensic Search Mechanism: Proof of Concept," *The Journal of Digital Forensics, Security and Law*, vol. 1, no. 1, pp. 25–44, 2006. [Online]. Available: http://commons.erau.edu/jdfsl/vol1/iss1/3/

46. D. Dumetz Carry, "Visual literacy: Using images to increase comprehension increase comprehension students need visual images to help them read and understand children live in a very," *Readingrecovery.org*, 2011.

47. W. Josh, "Why JSON Is Better Than XML," *API Industry Trends*, [Online]. Available: https://blog.cloud-elements.com/json-better-xml.[Accessed: Nov 12, 2018].

48. A. Dix, "Human computer interaction (hci)," *Interaction Design Foundation*, [Online] https://www.interaction-design.org/literature/topics/human-computer-interaction [Accessed: Jan 13, 2018].

49. H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale, "Empirical studies in information visualization: Seven scenarios," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 9, pp. 1520–1536, 2012.

50. P. Bartels, T. De Buyser, and J. Van Ussel, "A Football Data visualization : The Belgian First Division," *Department of Computer Science, KU Leuven, Belgium*, pp. 1–8, 2013.

51. P. Saraiya, C. North, and K. Duca, "An Evaluation of Microarray Visualization Tools for Biological Insight," *IEEE Symposium on Information Visualization*, pp. 1–8, 2004.

52. D. J. Schelkoph, "Digital Forensics Event Graph Reconstruction," *Ph.D. thesis, Department of the Air Force Air University, Air Force Institure of Technology, Wright-Patterson Air Force Base, Ohio*, 2016.

53. GraphQL, "A query language for your api." [Online]. Available: https://graphql.org/ [Accessed: Jan. 05, 2019].

54. J. Helfer, "Graphql explained-how graphql turns a query into a response."

55. "Caching data-a guide to customizing and directly accessing your apollo cache," *Apollo Client*, [Online]. Available: https://www.apollographql.com/docs/react/advanced/caching.html [Accessed: Jan. 04, 2019].

56. "Vis.js," *Almende B.V.*, [Online]. Available: http://visjs.org/docs/timeline/ [Accessed: Jan. 05, 2019].

57. R. Chandrawanshi and H. Gupta, "A survey: Server timeline analysis for web forensics," *International Journal of Scientific Research Engineering & Technology (IJSRET)*, vol. 1, no. 12, 2013.

58. H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale, "Seven Guiding Scenarios for Information Visualization Evaluation Seven Guiding Scenarios for Information Visualization Evaluation," *Technical Report: DCS University of Calgary*, p. 17, 2011.

59. National Institute of Standards and Technology (NIST), "Hacking case," [Online]. https://www.cfreds.nist.gov/Hacking_Case.html [Accessed: Nov. 05, 2018].

60. J. R. Lewis, "Psychometric evaluation of the post-study system usability questionnaire: The pssuq," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 36, no. 16, pp. 1259–1260, 1992. [Online]. Available: http://journals.sagepub.com/doi/10.1177/154193129203601617

61. M. M. Al-Zahrani, A. Nanni, S. U. Al-Dulaijan, and C. E. Bakis, "A Technique For the Measurement of Attitudes," *Proceedings of the Second International Conference on Advanced Composite Materials in Bridges and Structures (ACMBS-II)*, pp. 853–860, 1996.

62. J. Lewis, "Psychometric evaluation of the pssuq using data from five years of usability studies," *Int. J. Hum. Comput. Interaction*, vol. 14, pp. 463–488, 09 2002.

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704–0188*



| 1. REPORT DATE *(DD–MM–YYYY)* | 2. REPORT TYPE | 3. DATES COVERED *(From — To)* |
|---|---|---|
| 03–21–2019 | Master's Thesis | Sept 2017 — Mar 2019 |

**4. TITLE AND SUBTITLE**

Graph-based Temporal Analysis
in Digital Forensics

**5a. CONTRACT NUMBER**

**5b. GRANT NUMBER**

**5c. PROGRAM ELEMENT NUMBER**

**6. AUTHOR(S)**

Adderley, Nikolai, A 1stLt, USAF

**5d. PROJECT NUMBER**

**5e. TASK NUMBER**

**5f. WORK UNIT NUMBER**

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

Air Force Institute of Technology
Graduate School of Engineering an Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

**8. PERFORMING ORGANIZATION REPORT NUMBER**

AFIT-ENG-MS-19-M-005

**9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

Department of Defense Cyber Crime Center (DC3)
911 Elkridge Landing Rd
Linthicum Heights, MD 21090
POC: Eoghan Casey
Email: eoghan.casey@dc3.mil

**10. SPONSOR/MONITOR'S ACRONYM(S)**

DC3/DCCI

**11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

**12. DISTRIBUTION / AVAILABILITY STATEMENT**

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

**13. SUPPLEMENTARY NOTES**



**14. ABSTRACT**

Establishing a timeline as part of a digital forensics investigation is a vital part of understanding the order in which system events occurred. However, most digital forensics tools present timelines as histogram or as raw artifacts. Consequently, digital forensics examiners are forced to rely on manual, labor-intensive practices to reconstruct system events. Current digital forensics analysis tools are at their technological limit with the increasing storage and complexity of data. A graph-based timeline can present digital forensics evidence in a structure that can be immediately understood and effortlessly focused. This paper presents the Temporal Analysis Integration Management Application (TAIMA) to enhance digital forensics analysis via information visualization (infovis) techniques. TAIMA is a prototype application that provides a graph-based timeline for event reconstruction using abstraction and visualization techniques. A workflow illustration and pilot usability study provided evidence that TAIMA assisted digital forensics specialists in identifying key system events during digital forensics analysis.

**15. SUBJECT TERMS**

digital forensic, temporal analysis, forensic analysis, graph database, labeled property graphs, temporal event abstraction

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT | b. ABSTRACT | c. THIS PAGE | | | Dr. Gilbert Peterson |
| U | U | U | UU | 97 | 19b. TELEPHONE NUMBER *(include area code)* (937) 255-3636; gilbert.peterson@afit.edu |